Lezione 13

Gestione della memoria secondaria

Sistemi operativi

12 giugno 2012

Marco Cesati

System Programming Research Group
Università degli Studi di Roma Tor Vergata

Di cosa parliamo in questa lezione?

La gestione della memoria secondaria

1. I file system annotati
2. Tecnologia e prestazioni del disco magnetico
3. Algoritmi di schedulazione del disco
4. I sistemi RAID

Gestione della memoria secondaria
Marco Cesati

Schema della lezione
File system annotati
Il disco magnetico
Prestazioni dei dischi
Schedulazione del disco
Sistemi RAID

Ripristino di un file system

Il file system memorizza i file in un disco o una partizione per mezzo di strutture di dati registrate all'interno del disco stesso

Frequentemente, a causa di blackout o utilizzi impropri, un calcolatore viene spento senza smontare i file system in uso, causando

1. Perdita dei dati (scritture su file ancora in memoria cache e non avviate, oppure scritture in corso interrotte a metà)
2. Mancata cancellazione o creazione di file (modifiche di directory non avviate o incomplete)
3. Stato inconsistente del file system

I primi due problemi non sono rimediabili: l'informazione è andata persa prima che potesse essere salvata in una memoria persistente

Per ripristinare uno stato consistente il SO esegue una verifica del file system al riavvio successivo (*file system check*)

File system annotati

Il ripristino del file system:

- può durare ore per un disco da migliaia di gigabyte
- allunga i tempi di "fuori servizio" dei sistemi

I *file system annotati* (*journaling file system*) permettono di evitare la scansione di tutto il file system in caso di blackout

- Tutte le modifiche sui metadati sono annotate in una zona del disco chiamata log o journal
- Ogni insieme di operazioni che esegue uno specifico compito è annotata come transazione sul log **prima** di eseguire l'operazione sui file reali
- In caso di riavvio dopo un blackout, il file system esamina il proprio log e esegue tutte le operazioni corrispondenti alle transazioni complete presenti

Si possono perdere dati in caso di blackout? ***Sì!***

La presenza del log non garantisce che i dati siano scritti in memoria persistente prima dello spegnimento

Gestione della memoria secondaria
Marco Cesati

Schema della lezione
File system annotati
Il disco magnetico
Prestazioni dei dischi
Schedulazione del disco
Sistemi RAID

File system annotati (2)

Perché i file system annotati evitano la corruzione?

- Dal punto di vista del SO, il trasferimento di un blocco è atomico: o l'operazione ha successo, oppure fallisce
  - Molti dischi rigidi hanno batterie tampone che in caso di blackout consentono di concludere le scritture iniziate e "parcheggiare" le testine
- La corruzione del file system può avvenire in caso di operazioni che coinvolgono la modifica di più blocchi
  - Ad es. per cancellare un file si deve aggiornare una directory, marcare il FCB come disponibile, e marcare alcuni blocchi di dati come liberi
- Il log file contiene transazioni, ossia una sequenza di scritture che debbono essere eseguite in modo atomico
- In fase di riavvio il file system (ri-)esegue rapidamente tutte le operazioni corrispondenti a transazioni nel log marcate come chiuse e non completate

Il SO deve garantire che le modifiche del log siano scritte sul disco prima di avviare le operazioni corrispondenti

File system annotati (3)

I file system annotati possono anche essere utilizzati per evitare la corruzione dei dati entro un file (non solo metadati)

Tre livelli di annotazione disponibili in Ext3 (Linux):

- Writeback: solo operazioni su metadati trascritte nel log
  - Evita soltanto la corruzione dei metadati
- Ordered: solo operazioni su metadati sono trascritte nel log; tuttavia è garantito che una transazione nel log è marcata come completata solo dopo la scrittura dei dati
  - Evita la corruzione dei dati per scritture in coda al file; non evita la corruzione per scritture in mezzo ai file
- Journal: anche le operazioni di modifica dei dati sono trascritte nel log
  - Evita teoricamente la corruzione dei dati
  - Ogni blocco è scritto due volte sul disco

La possibilità di evitare la corruzione è basata sull'assunzione che il disco completi sempre le scritture iniziate e nell'ordine di sottomissione: in pratica ciò non è necessariamente vero

La memoria secondaria

La *memoria secondaria* di un calcolatore è costituita da dispositivi di memorizzazione:

- Persistenti (conservano i dati anche senza alimentazione)
- Di grande capacità (ordine di centinaia di miliardi di byte)
- Di costo contenuto

Esistono molti tipi di dispositivi per memoria secondaria, che differiscono per costo, velocità e capacità

Tra i più importanti oggi:

- I dischi magnetici
- I nastri magnetici
- I dischi ottici
- I dischi a stato solido

Il disco magnetico

Il *disco magnetico* (o *disco rigido*) è il dispositivo di memoria secondaria più diffuso, in quanto:

- è relativamente veloce
- è poco ingombrante
- può avere capacità elevata
- è abbastanza affidabile

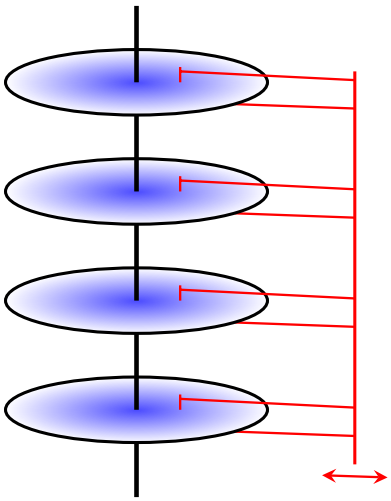
Il disco magnetico è costituito da una schiera di piatti rotanti rivestiti di un sottile strato di materiale ferromagnetico (sovente su entrambe le faccie)

Una *testina di lettura/scrittura* è sospesa pochi micron sopra ciascun piatto

Le testine di ciascun piatto sono montate su un unico perno e si spostano lungo la direzione radiale dei piatti in modo solidale

I piatti ruotano a velocità elevate e costanti, tipicamente dell'ordine di 5 000–15 000 RPM (giri al minuto)

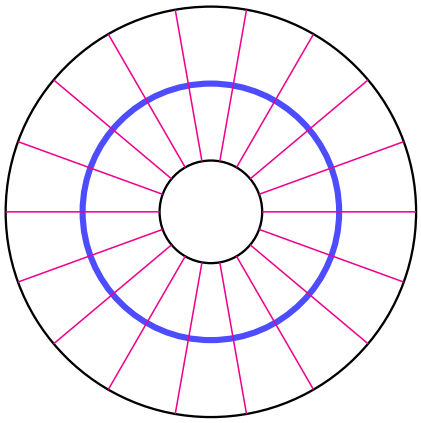
Il disco magnetico (2)

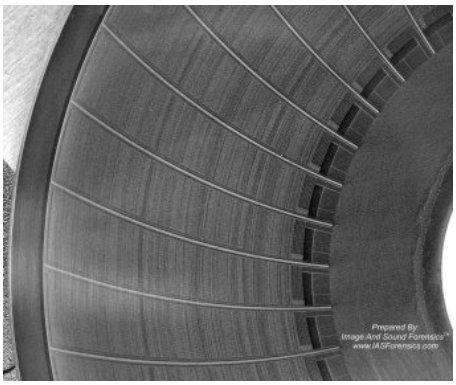


Organizzazione dei dati

- La superficie di ciascun piatto è suddivisa logicamente in anelli concentrici chiamati *tracce*
- Sullo stesso piatto possono essere presenti da 10 000 a 50 000 tracce, opportunamente distanziate tra loro
- Ogni traccia corrisponde ad una possibile posizione radiale della testina di lettura/scrittura
- Ogni traccia memorizza una sequenza di bit sotto forma di stati di magnetizzazione della superficie del piatto
- La rotazione del piatto fa scorrere la traccia sotto alla testina ed induce una corrente che rileva il valore dei bit
- Ogni traccia è suddivisa in qualche centinaio di *settori*, ciascuno dei quali memorizza in genere 512 byte di dati
- L'insieme di tutte le tracce lette contemporaneamente dalle testine è chiamato *cilindro*

Organizzazione dei dati (2)





Come si può identificare un dato all'interno del disco?

È necessario indicare: il numero di cilindro, il numero di testina, ed il numero di settore nella traccia

Come si determina la posizione corrente delle testine?

- Il cilindro sotto la testina è correlato alla posizione del perno
- Dentro il settore è memorizzato anche il numero progressivo del settore stesso entro la traccia

Prestazioni dei dischi magnetici

Le principali misure di prestazione dei dischi magnetici sono:

- La latenza delle letture e scritture dei settori
- La velocità di trasferimento massima di settori consecutivi

Quale è il parametro più importante per i dischi magnetici?

È il tempo di latenza delle letture e scritture, in quanto in generale i dati dei dischi rigidi non sono acceduti in sequenza rispetto alla loro posizione fisica sui piatti

Il *tempo di latenza* delle letture e scritture è costituito da:

- Il *tempo di posizionamento* (*seek*) necessario per muovere le testine sul cilindro corretto
- Il *tempo di rotazione* necessario per far arrivare sotto la testina appropriata il settore da trasferire
- Il *tempo di trasferimento* necessario per leggere tutti i bit del settore
- Il *ritardo del controller* introdotto dai circuiti elettronici che gestiscono e interfacciano la schiera di piatti

Tempo di latenza di un disco magnetico

Un tipico disco magnetico ha:

- Settore: 512 byte
- Ritardo del controller: 200 μsec
- Velocità di trasferimento: 50 MB/sec
- Velocità di rotazione: 10 000 RPM
- Tempo medio di posizionamento: 6 msec

Qual è la latenza media di una operazione su un settore?

Ciascun piatto compie un giro in 60/10 000 sec=6 msec, quindi il tempo di rotazione è in media 3 msec. Di conseguenza, la latenza media è $0{,}2 + 6 + 3 + (512 / 52\,428\,800) = 9{,}2$ msec

Conviene utilizzare dischi con velocità di rotazione maggiori?

Solo fino ad un certo punto: ad esempio, lo stesso disco con una velocità di rotazione di 15 000 RPM (+50%) avrebbe una latenza media di 8,2 msec (-10%)

Il vero guadagno sta nel ridurre il *tempo di posizionamento*!

Ridurre il tempo di posizionamento

Le prestazioni di un disco magnetico dipendono soprattutto dal tempo di posizionamento delle testine sul cilindro desiderato

Come si può ridurre il tempo di posizionamento?

- Via hardware: introdurre più testine di lettura e scrittura sullo stesso piatto, in modo da ridurre il tempo di posizionamento (ed anche il tempo di rotazione)
- Via software (SO): accorpare diverse richieste di lettura o scrittura di settori contigui in una singola operazione che possa essere servita con una singola operazione di posizionamento

I dischi magnetici moderni fanno anche uso di una memoria RAM a bordo del controller per mitigare gli effetti dei ritardi di posizionamento e rotazione

Gestione della memoria secondaria
Marco Cesati

Schema della lezione
File system annotati
Il disco magnetico
Prestazioni dei dischi
Schedulazione del disco
Sistemi RAID

Schedulazione del disco

Le richieste di trasferimento di dati da un disco magnetico possono essere gestite tramite una struttura di dati dinamica chiamata *coda di I/O*

Ciascun elemento della coda di I/O di un disco contiene

- il numero di settore da trasferire
- il tipo di trasferimento (lettura o scrittura)
- l'indirizzo di memoria centrale per i dati

Poiché i trasferimenti sono molto più lenti delle operazioni delle CPU e della memoria centrale, la coda di I/O può contenere un gran numero di richieste in attesa

L'*algoritmo di schedulazione del disco* definisce l'ordine in cui il SO evade le richieste nella coda di I/O

La scelta dell'algoritmo di schedulazione è critica per le prestazioni del sistema

La scelta di un algoritmo di schedulazione ottimale richiede la conoscenza della *geometria fisica* del disco

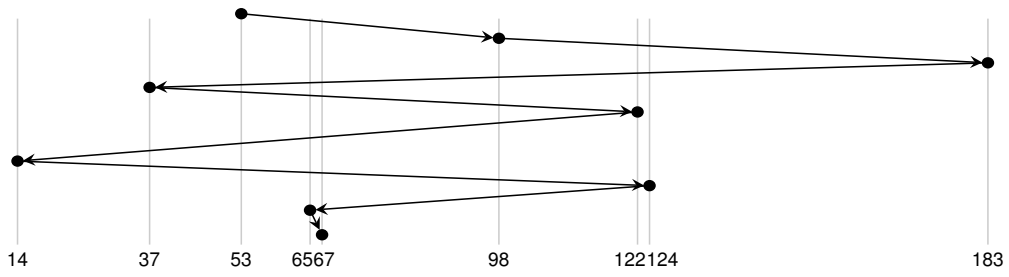
Schedulazione FCFS

La schedulazione in ordine d'arrivo FCFS (First Come, First Served) evade le richieste di trasferimento privilegiando quella che attende in coda da più tempo

Quali sono vantaggi e svantaggi di questo algoritmo?

- Vantaggio: la schedulazione è intrinsecamente equa, e nessuna richiesta può essere trascurata
- Svantaggio: le prestazioni possono essere cattive a cause dei tempi di posizionamento delle testine

Testina su 53; richieste in coda: 98, 183, 37, 122, 14, 124, 65, 67



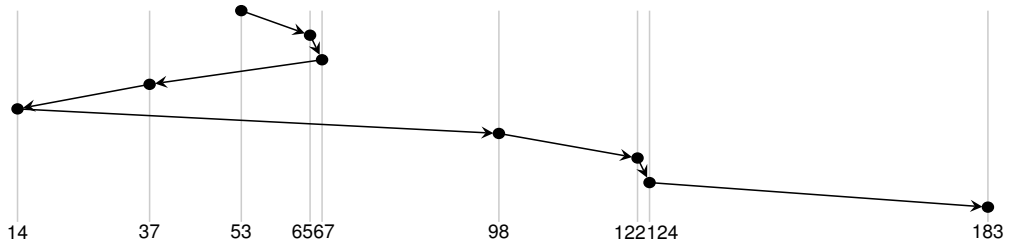
Schedulazione SSTF

La schedulazione per brevità SSTF (Shortest Seek Time First) evade la richiesta con il minimo tempo di riposizionamento rispetto alla posizione corrente delle testine

Quali sono vantaggi e svantaggi di questo algoritmo?

- Vantaggio: i ritardi dovuti ai tempi di riposizionamento sono fortemente ridotti
- Svantaggio: non è equo, alcune richieste possono attendere indefinitivamente

Testina su 53; richieste in coda: 98, 183, 37, 122, 14, 124, 65, 67



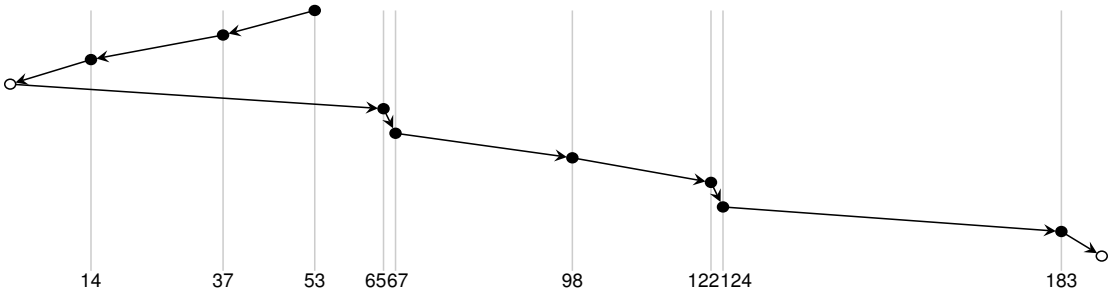
Schedulazione SCAN

La schedulazione per scansione SCAN (o algoritmo ascensore) evade le richieste muovendo le testine in entrambe le direzioni da un estremo all'altro del disco

Quali sono vantaggi e svantaggi di questo algoritmo?

- Vantaggio: più equo di SSTF
- Svantaggi: movimenti inutili in assenza di richieste; quando la testina è ad un estremo, le richieste all'altro estremo sono "vecchie" ma saranno soddisfatte per ultime

Testina su 53 (←); richieste in coda: 98, 183, 37, 122, 14, 124, 65, 67



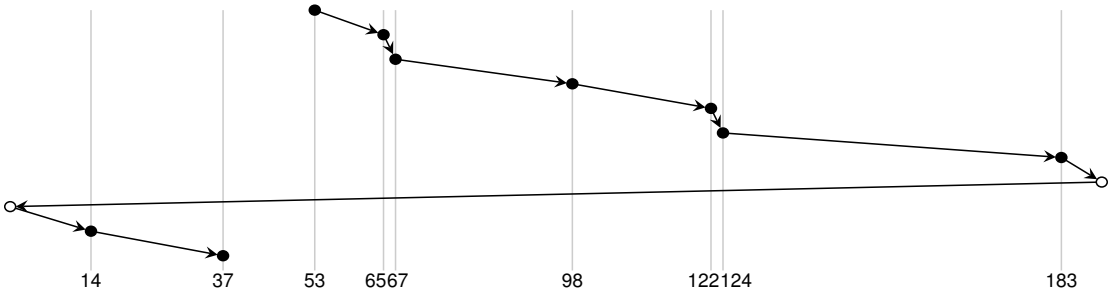
Schedulazione C-SCAN

La schedulazione per scansione circolare C-SCAN evade le richieste muovendo le testine in una unica direzione da un estremo all'altro del disco

Quali sono vantaggi e svantaggi di questo algoritmo?

- Vantaggio: più equo di SCAN, con meno variabilità nei tempi d'attesa
- Svantaggi: movimenti inutili in assenza di richieste

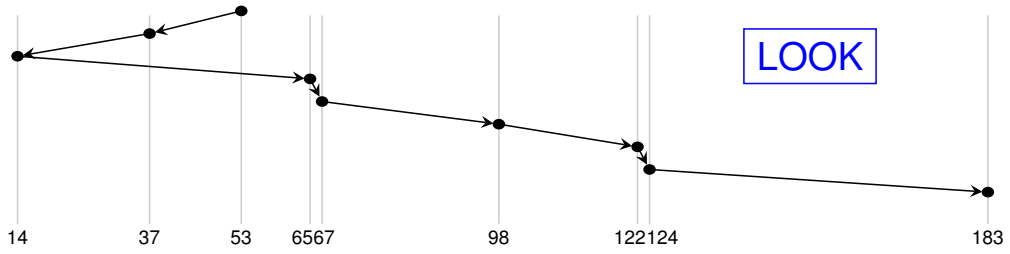
Testina su 53 (→); richieste in coda: 98, 183, 37, 122, 14, 124, 65, 67



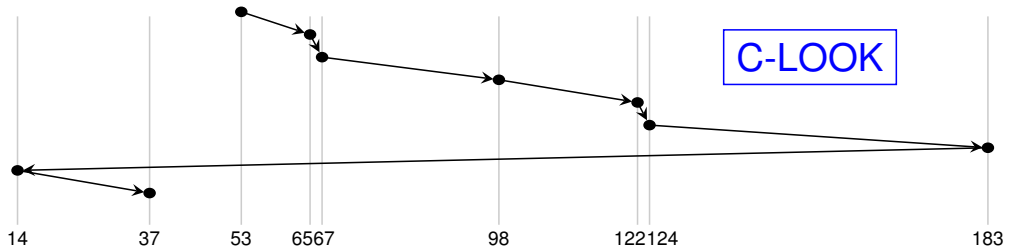
Schedulazioni LOOK e C-LOOK

Le schedulazioni LOOK e C-LOOK sono varianti delle schedulazioni SCAN e C-SCAN: quando invertono la direzione le testine "osservano" le richieste in attesa e non arrivano necessariamente agli estremi del disco

Testina su 53 (←); richieste in coda: 98, 183, 37, 122, 14, 124, 65, 67



Testina su 53 (→); richieste in coda: 98, 183, 37, 122, 14, 124, 65, 67



RAID

- Le prestazioni dei dischi magnetici crescono del 25%–50% all'anno: diventano sempre più lenti rispetto ai processori
- Anche la loro affidabilità non cresce in modo commisurato all'aumento di capacità
- Idea: usare più dischi magnetici in parallelo per
  - aumentare la velocità di funzionamento
  - aumentare la tolleranza ai guasti

Si definisce sistema RAID (Redundant Array of Inexpensive (o Independent) Disks) un insieme di dischi gestito in modo coordinato da un unico controllore (hardware o software)

Sono stati definiti diversi standard RAID, ciascuno con diversi livelli di costo, efficienza e affidabilità

RAID 0 (nessuna ridondanza)

- I dati vengono suddivisi su più dischi in modo automatico
- Si effettua lo striping: blocchi di dati logicamente sequenziali (chiamati *strisce*) vengono memorizzati su dischi differenti
- Un accesso che coinvolge due o più strisce consecutive (ad esempio, lettura sequenziale di un file) può essere effettuato trasferendo i dati in parallelo su più dischi
- Non esiste alcuna ridondanza dei dati, ed in effetti non è un vero e proprio sistema RAID
- Spesso utilizzato per elaborazioni con grandi quantità di dati in cui singoli errori sono tollerabili (ad esempio, elaborazioni di video digitali)

Disco 0	B0	B4	B8	...
Disco 1	B1	B5	B9	...
Disco 2	B2	B6	B10	...
Disco 3	B3	B7	B11	...

RAID 1 (mirror)

- Nel RAID 1 (detto anche *mirroring* o *shadowing*) ogni striscia è duplicata su più dischi
- Ogni disco è duplicato su un intero disco che funziona come copia di sicurezza
- Permette di avere buone prestazioni in lettura, ma ogni scrittura deve essere duplicata
- È la soluzione più costosa perché richiede il doppio dei dischi a parità di capacità
- Utilizzato per dati e file altamente critici, in quanto garantisce un backup in tempo reale immediatamente disponibile in caso di guasto
- Spesso utilizza come dischi dei sistemi RAID 0 (RAID 0+1)

Disco 0	B0	B1	B2	...
Disco 1	B0	B1	B2	...

RAID 2 e RAID 3

Il sistema RAID 1 è costoso, e il sistema RAID 0 è vulnerabile ai guasti: come potrebbe essere fatto un sistema RAID migliore?

L'idea è quella di utilizzare codici di correzione/rilevazione degli errori al posto delle copie complete dei dati

- RAID 2 e RAID 3 usano tecniche di accesso parallelo: tutti i dischi sono gestiti in modo coordinato in modo da avere sempre lo stesso cilindro sotto le testine
- In ogni operazione sono coinvolti tutti i dischi dell'array
- I dati si distribuiscono (striping) bit a bit (strisce uguali a parole o byte)
- Consentono velocità di trasferimento elevate
- Hanno tempi di latenza degli accessi elevato

RAID 2 (ridondanza tramite codice di Hamming)

- Utilizza un codice per correzione d'errore (codice di Hamming)
- I bit di ciascuna parola sono distribuiti su alcuni dischi, ed i loro bit di controllo sono su altri dischi
- I bit di controllo possono correggere un singolo errore o ricostruire il contenuto di un disco rotto
- Richiede molti dischi: ad es., 32 dischi per i bit di una parola, 7 dischi per i bit di controllo
- È costoso anche per la complessità del controllore
- Utilizzato molto raramente

Disco 0	b_0	b_4	b_8	...
Disco 1	b_1	b_5	b_9	...
Disco 2	b_2	b_6	b_{10}	...
Disco 3	b_3	b_7	b_{11}	...
Disco 4	p_0	p_3	p_6	...
Disco 5	p_1	p_4	p_7	...
Disco 6	p_2	p_5	p_8	...

RAID 3 (parità con bit interlacciato)

- Versione semplificata di RAID 2: utilizza bit di parità al posto del codice di Hamming
- Un solo disco aggiuntivo per i bit di parità
- Non consente di correggere errori singoli, ma permette di ricostruire il contenuto di un disco rotto
- Alti tempi di latenza di accesso ai dati, ma grandi velocità di trasferimento (tutti i dischi lavorano sempre in parallelo)
- Utilizzato per elaborazioni con grandi quantità di dati

Disco 0	b_0	b_4	b_8	...
Disco 1	b_1	b_5	b_9	...
Disco 2	b_2	b_6	b_{10}	...
Disco 3	b_3	b_7	b_{11}	...
Disco 4	p_0	p_1	p_2	...

0	1	0	0	...
1	1	0	0	...
1	0	0	0	...
0	1	1	0	...
0	1	1	0	...

RAID 4, RAID 5 e RAID 6

RAID 2 e RAID 3 hanno alti tempi di latenza perché soddisfano una sola richiesta di trasferimento alla volta: come si può realizzare un sistema RAID per abbassare i tempi di latenza?

È necessario far lavorare i dischi in modo indipendente!

- Negli standard RAID 4, RAID 5 e RAID 6 i dischi operano in modo indipendente
- Lo striping è fatto a blocchi, con strisce di decine di KB
- I bit di parità sono memorizzati in blocchi, ciascuno dei quali è relativo ad un insieme di blocchi
- Permettono di effettuare *accessi piccoli*, ossia operazioni che coinvolgono un singolo disco
- È possibile effettuare più accessi piccoli in parallelo
- Rispetto a RAID 2 e RAID 3, hanno tempi di latenza inferiori, ma anche minori velocità di trasferimento

RAID 4 (parità a livello di blocco)

- Striping a livello di singoli blocchi (come RAID 0)
- Un disco contiene tutti i blocchi di parità relativi ai corrispondenti blocchi negli altri dischi
- I blocchi di parità sono utilizzati solo per ricostruire il contenuto di un disco rotto
- Consente la lettura di blocchi indipendenti in parallelo su più dischi
- Nelle operazioni di scrittura il disco con i blocchi di parità è un *collo di bottiglia*

Disco 0	B0	B4	B8	...
Disco 1	B1	B5	B9	...
Disco 2	B2	B6	B10	...
Disco 3	B3	B7	B11	...
Disco 4	P0-3	P4–7	P8–11	...

Operazione di scrittura in RAID 4

Ciascun blocco di parità in RAID 4 è l'XOR di un certo numero N di blocchi di dati in altrettanti dischi

Per ogni operazione di scrittura di un blocco dati si dovrebbe:

- Scrivere il nuovo blocco dati
- Leggere gli $N - 1$ blocchi dati nello stesso gruppo
- Calcolare i nuovi bit di parità e scrivere il blocco di parità

Totale: 2 scritture e $N - 1$ letture

E' possibile implementare meglio l'operazione di scrittura? ***Sì!***

- Leggere il blocco di parità P
- Leggere il contenuto del blocco di dati da sovrascrivere B
- Scrivere il blocco di dati B'
- Calcolare i nuovi bit di parità come $P' = P \oplus B \oplus B'$ e scrivere il blocco di parità P'

Totale: 2 scritture e 2 letture

RAID 5 (parità distribuita a livello di blocco)

- Simile a RAID 4, con la differenza che i blocchi di parità sono distribuiti su tutti i dischi invece che memorizzati in un disco dedicato
- Generalmente la distribuzione dei blocchi di parità è in modalità round-robin (a rotazione)
- La logica di controllo è un po' più complessa
- Il grande vantaggio è che non esiste più un unico collo di bottiglia per le operazioni di scrittura
- Sono anche possibili operazioni di scrittura parallele se coinvolgono dischi differenti (nell'es., B5 e B11)

Disco 0	B0	B4	B8	B12	...
Disco 1	B1	B5	B9	P12–15	...
Disco 2	B2	B6	P8–11	B13	...
Disco 3	B3	P4–7	B10	B14	...
Disco 4	P0-3	B7	B11	B15	...

RAID 6 (ridondanza duale)

- Ai blocchi di parità (blocchi P) si aggiungono altri blocchi di controllo (blocchi Q) che utilizzano un algoritmo differente
- L'aggiunta dei blocchi di controllo Q permette di rimediare ad un guasto aggiuntivo: i dati si preservano anche in presenza di due dischi rotti
- Utilizzato in sistemi che devono preservare i dati con altissime probabilità
- Le operazioni di scrittura sono più lente rispetto a RAID 5

Disco 0	B0	B4	B8	B12	...
Disco 1	B1	B5	B9	$P_{12–15}$	...
Disco 2	B2	B6	$P_{8–11}$	$Q_{12–15}$	...
Disco 3	B3	$P_{4–7}$	Q_{8-11}	B13	...
Disco 4	P_{0-3}	Q_{4-7}	B10	B14	...
Disco 5	Q_{0-3}	B7	B11	B15	...

Configurazioni RAID annidate

I dispositivi RAID possono anche essere utilizzati in modo "annidato"

Generalmente, uno degli standard sempre utilizzati è RAID 0: infatti l'annidamento serve essenzialmente a migliorare le prestazioni del sistema di memorizzazione

La configurazione nota come RAID 0+1 è una duplicazione di *striscie* (*striped mirror*)

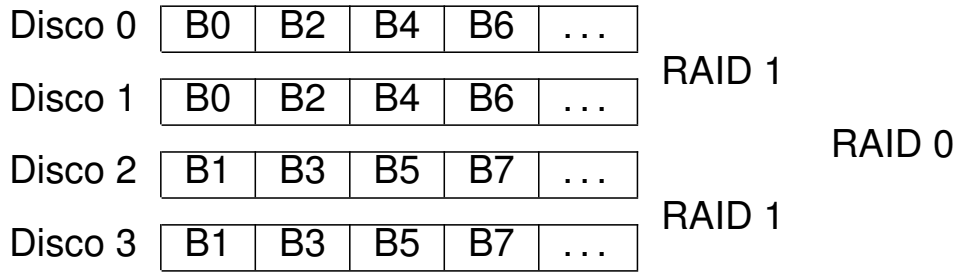
Disco 0	B0	B2	B4	B6	...	RAID 0
Disco 1	B1	B3	B5	B7	...	
Disco 2	B0	B2	B4	B6	...	RAID 0
Disco 3	B1	B3	B5	B7	...	

RAID 1

Configurazioni RAID annidate (2)

La configurazione RAID 1+0 (*stripe of mirrors*) è preferibile a RAID 0+1 perché più resistente ai guasti

Poiché la configurazione dei blocchi sui dischi è identica, esistono sistemi RAID in grado di convertire la modalità di funzionamento da RAID 1+0 a RAID 0+1 e viceversa secondo necessità



Vengono utilizzate molte altre configurazioni annidate; ad esempio: RAID 0+3, RAID 3+0, RAID (1+0)+0, RAID 5+0, RAID 6+0, . . .