



 POLITECNICO DI MILANO

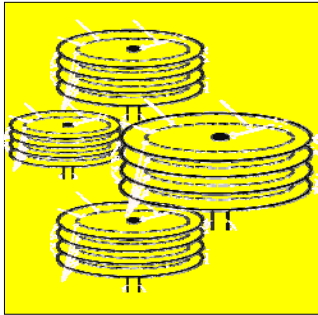
Impianti Informatici



RAID

➔
Redundant Arrays of Independent (Inexpensive) Disks
2

Redundant Arrays of Independent (Inexpensive) Disks



Parallelismo

Elevato Transfer Rate

- I/O pesanti

Elevato I/O Rate

- I/O leggeri

Load balancing

"A case for Redundant Arrays of Inexpensive Disks (RAID)" - D. Patterson, 1988

Impianti Informatici
POLITECNICO DI MILANO

1. Il RAID è un sistema che usa molti dischi indipendenti,
2. visti però come un unico grande disco logico ad elevate prestazioni.
3. I dati sono distribuiti su più dischi che sono acceduti in parallelo, permettendo in questo modo:
4. Un elevato transfer rate quando si ha a che fare con
5. operazioni di I/O particolarmente pesanti, cioè in cui si deve trasferire una grande quantità di dati
6. Un elevato I/O rate per operazioni di I/O
7. leggere, cioè con accessi a piccole quantità di dati
8. Load balancing automatico dei dischi
9. L'originario acronimo di RAID indicava un insieme ridondante di dischi economici, sottolineando così il suo uso nel combinare assieme molti dischi poco costosi e obsoleti.
10. La definizione attuale si riferisce invece ad un insieme di dischi indipendenti.

➔
Ottimizzazione delle prestazioni: Read Caching
3


Read Caching

Prefetching

Write Buffering

I dati più probabili vengono mantenuti in una memoria cache

- Dati: blocchi di 4 Kbyte
- Metodo di alimentazione LRU.



I dischi hanno spesso una loro cache

Read miss ratio

- Frazione di operazioni che richiedono l'accesso al disco
- Dato non in cache
- Indica l'efficienza
- Dimensione della cache: ottimale fino al 4% dello storage totale

Impianti Informatici
POLITECNICO DI MILANO

1. Il Read Caching è una tecnica che consiste nel tenere temporaneamente
 2. in una memoria più veloce, una *cache* appunto,
 3. quei dati che si pensa sia probabile vengano usati.
 4. Per sfruttare il principio della località spaziale, i dati vengono memorizzati a blocchi, con dimensione tipiche di 4kbyte.
 5. Molti dischi implementano al loro interno una memoria cache, spesso usata, come vedremo in seguito, come buffer di prefetching.
- Si definisce *read miss ratio*
 - la frazione di operazioni che richiedono l'operazione fisica sul disco,
 - e che quindi non giovano della cache.
 - Esso è un buon indice dell'efficienza della cache; tale indicatore migliora al crescere delle dimensioni della cache,
 - almeno fino a che essa raggiunge il 4% della memoria storage usata

Ottimizzazione delle prestazioni: Read Caching

Read Caching

Prefetching

Write Buffering

- L'analisi di dati sperimentali mostra una miss ratio che segue la relazione:

$$f(x) = a(x-b)^c \quad (a, b, c \text{ costanti, } c = -1)$$
- La relazione funzionale è simile a quella che si trova a livello logico per i buffer dei database (ma in questo caso il valore di c è circa la metà, in altre parole si trova che il caching a livello fisico è più efficiente di quello ottenuto a livello logico).

8%

dimensione della cache
(% dei dati)

POLITECNICO DI MILANO

1. Il Read Caching è una tecnica che consiste nel tenere temporaneamente
 2. in una memoria più veloce, una *cache* appunto,
 3. quei dati che si pensa sia probabile vengano usati.
 4. Per sfruttare il principio della località spaziale, i dati vengono memorizzati a blocchi, con dimensione tipiche di 4kbyte.
 5. Molti dischi implementano al loro interno una memoria cache, spesso usata, come vedremo in seguito, come buffer di prefetching.
- Si definisce *read miss ratio*
 - la frazione di operazioni che richiedono l'operazione fisica sul disco,
 - e che quindi non giovano della cache.
 - Esso è un buon indice dell'efficienza della cache; tale indicatore migliora al crescere delle dimensioni della cache,
 - almeno fino a che essa raggiunge il 4% della memoria storage usata

 **Ottimizzazione delle prestazioni: Prefetching** 5

Read Caching

Prefetching

Write Buffering

Precarica in memoria i dati che si presume saranno richiesti a breve

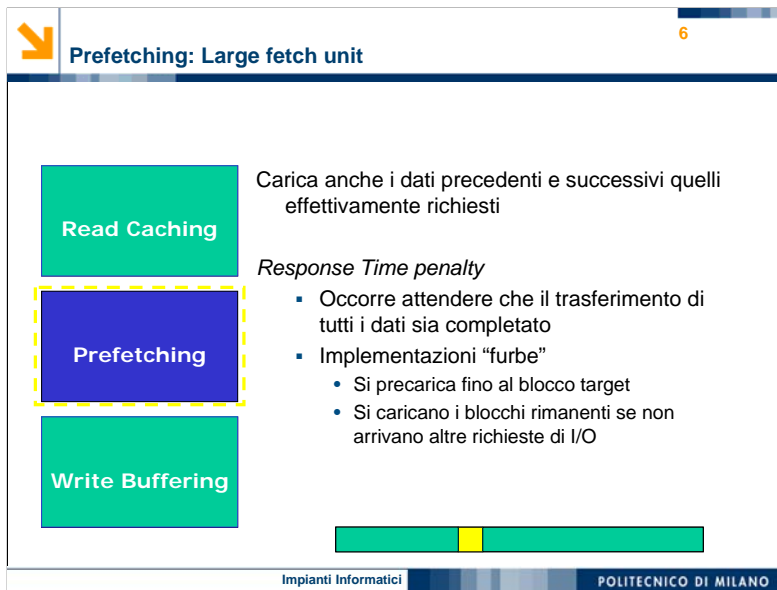
- Accuratezza previsione
- Costi di prefetching (risorse consumate)
- Tempestività dell'operazione (operazione completa prima che i dati servano)

Molti accessi al disco sono sequenziali

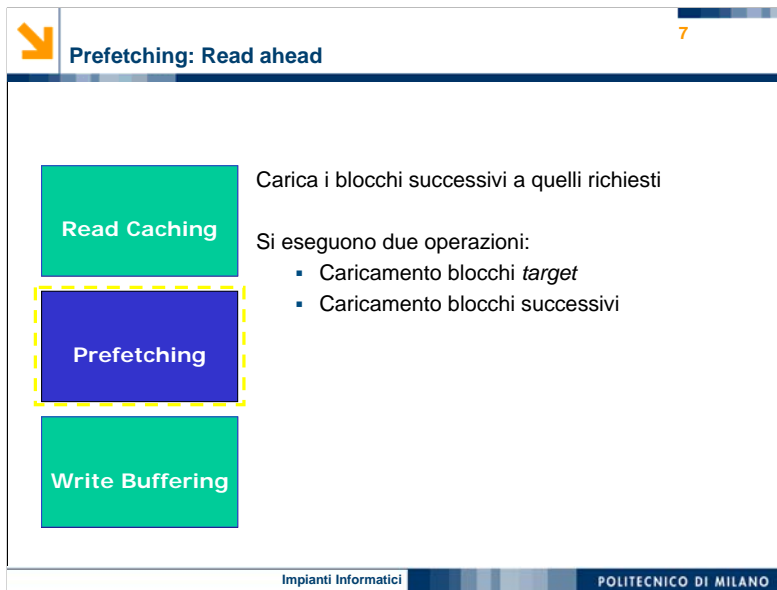
- Si può realizzare prefetching sequenziale in occasione di una *cache miss*
- Molteplici accessi vengono trasformati in uno singolo

Impianti Informatici POLITECNICO DI MILANO

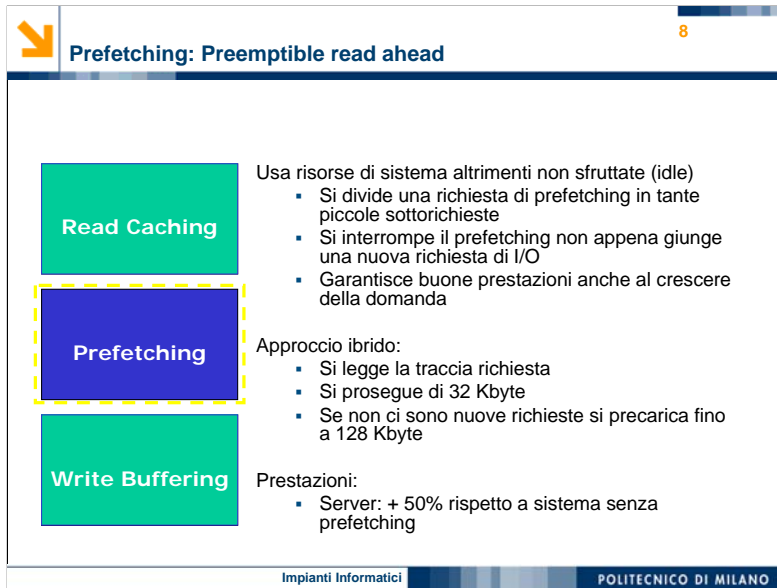
1. Il prefetching è un'ottimizzazione che consiste nel prevedere quali blocchi è probabile vengano usati in futuro e caricarli prima che siano effettivamente richiesti.
2. Occorre tener conto dell'accuratezza della previsione;
3. Del costo, in termini di risorse consumate, sia di memoria, cpu e altri componenti;
4. È inoltre importante la tempestività, cioè che l'operazione venga completata prima che i blocchi siano necessari
5. Una buona tecnica di prefetching si può basare sulla considerazione che la maggior parte dei carichi presenta una certa sequenzialità nella distribuzione degli accessi;
 - affiancata alla tecnica di caching, si può precaricare, in occasione di un *cache miss*, la sequenza di dati successivi.
 - Un prefetching sequenziale trasforma in un singolo I/O parecchi I/O di blocchi più piccoli



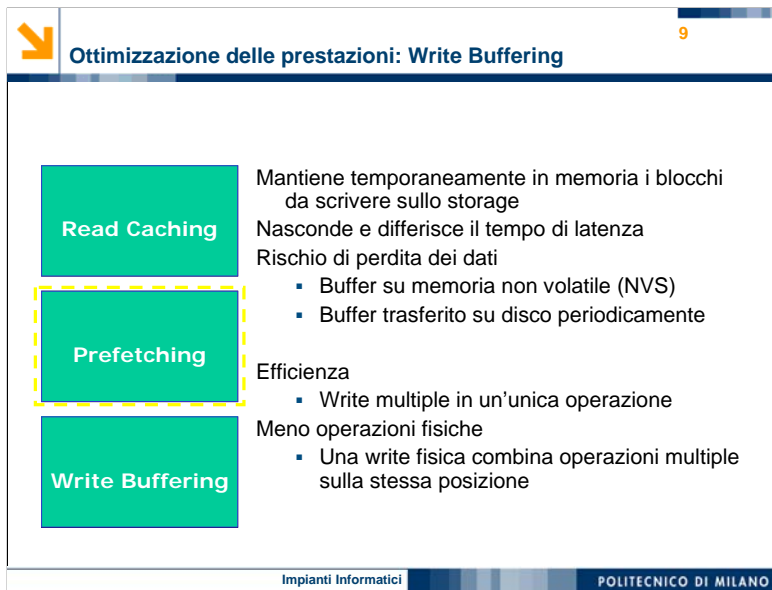
1. Mediante il Large Fetch unit, vengono caricati, in un'unica operazione, un certo numero di blocchi che precedono e che seguono quelli effettivamente richiesti
2. Lo svantaggio è che bisogna attendere che il trasferimento sia completato,
3. in alternativa si possono eseguire due operazioni separate per ridurre questa penalità:
4. Si esegue il prefetching dei dati fino al blocco “target”,
5. e poi dei blocchi rimanenti se non ci sono altre richieste di I/O



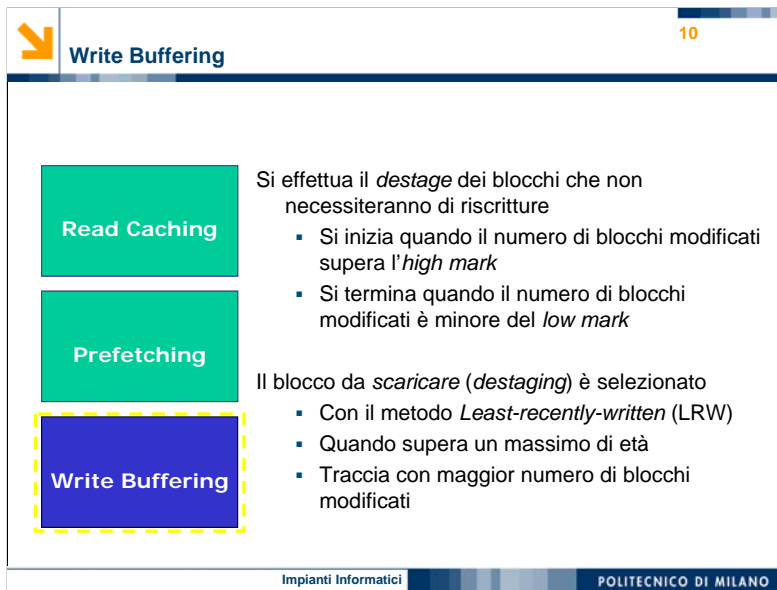
1. Tramite read ahead, dopo che i blocchi richiesti sono stati ottenuti, si prosegue a leggere in avanti e a precaricare tali dati.
- Generalmente vengono eseguite due operazioni distinte:
 - una per i blocchi “target”
 - e un'altra per quelli che seguono




1. Una semplice forma di *prefetch* che usa solo risorse che altrimenti sarebbero *idle* o non usate è il Preemptible read ahead
2. la richiesta è divisa in tante richieste sotto-richieste,
3. in modo che la sequenza è interrotta quando arriva una nuova richiesta.
4. In questo modo si evita che al crescere della domanda, cioè del numero di blocchi, le prestazioni comincino a degradare.
5. Il metodo migliore sembra essere l'approccio ibrido,
6. iniziando a leggere la traccia su cui si è posizionati,
7. e proseguendo di 32 KB oltre i blocchi richiesti; se poi
8. non ci sono nuove richieste si prosegue fino a 128 KB.
9. I miglioramenti di prestazione, nel caso dei server,
10. sono almeno del 50% rispetto ai sistemi senza prefetching.



1. Il Write buffering è una tecnica che consiste nel mantenere temporaneamente in memoria i blocchi scritti prima di fare un destaging dei dati nella storage permanente
- l'operazione di write è data come completata quando il dato è scritto nel buffer. In questo modo il tempo di latenza è nascosto e differito nel tempo.
 - per evitare perdita di dati il *write buffer*
 - è eseguito su memoria non volatile oppure,
 - in un approccio più economico, il contenuto dei buffer viene trasferito su disco periodicamente (ad es. ogni 30 secondi)
 - Con write buffering si ha una maggiore efficienza in scrittura,
 - perché write multiple consecutive sono combinate in una operazione.
 - Si riduce inoltre il numero di operazioni fisiche,
 - perché una singola write fisica combina operazioni multiple sulla stessa posizione.



1. Tramite il write buffering si tenta di ridurre il numero di write fisiche operando il *destage* dei blocchi su cui è meno probabile avvengano in futuro riscritture.
2. Il processo di *destage* del buffer inizia quando il numero di blocchi modificati è superiore a una soglia, detta *high mark*
3. e termina quando scende sotto il limite, denominato *low mark*;
4. il blocco da scaricare è selezionato
5. in base al metodo *least-recently-written*
6. o quando ha superato un massimo stabilito di età;
7. Può inoltre essere selezionata per il *destage* la traccia col maggior numero di blocchi modificati


Write Buffering
11

Read Caching

Prefetching

Write Buffering

Equilibrio tra eliminazione delle write e scrittura multipla

- *High mark* = 0.8
- *Low mark* = 0.2

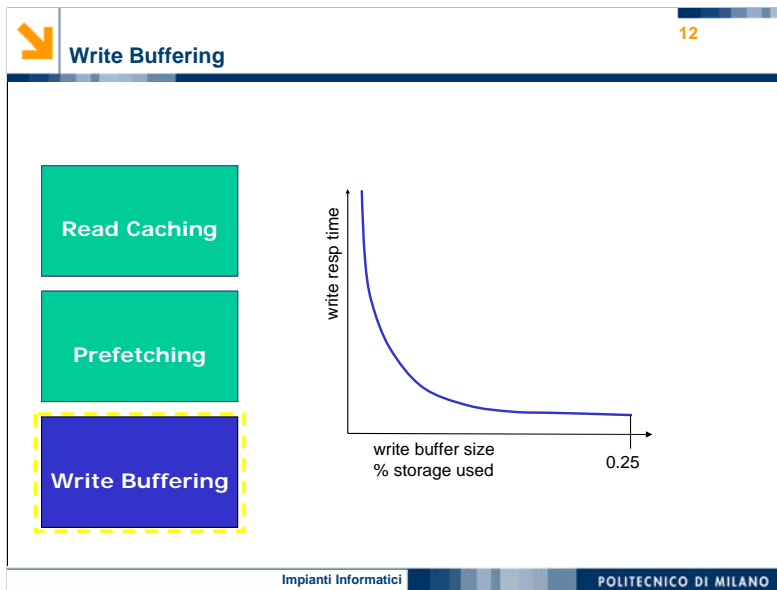
Se $Low\ mark \ll High\ mark$ il *destage* avviene a lotti

Per minimizzare il tempo di attesa si possono ordinare le write fisiche, in base a:

- Tempo minimo di accesso
- Tempo minimo di posizionamento

Impianti Informatici
POLITECNICO DI MILANO

1. l'eliminazione delle write ripetute e la scrittura multipla competono per spazio. L'equilibrio è raggiunto con un appropriato aggiustamento dei valori di soglia;
2. una buona efficienza si consegue con: *high mark* = 0.8
3. e *low mark* = 0.2
4. Se la soglia minima è considerevolmente inferiore a quella massima, le operazioni di *destage* avvengono in lotti;
5. le operazioni fisiche di write possono essere ordinate in modo da minimizzare il tempo di attesa
6. Ad esempio si possono selezionare le richieste da servire in base al minimo tempo di accesso, oppure
7. al minimo tempo di posizionamento stimato




1. l'eliminazione delle write ripetute e la scrittura multipla competono per spazio. L'equilibrio è raggiunto con un appropriato aggiustamento dei valori di soglia;
2. una buona efficienza si consegue con: *high mark* = 0.8
3. e *low mark* = 0.2
4. Se la soglia minima è considerevolmente inferiore a quella massima, le operazioni di destage avvengono in lotti;
5. le operazioni fisiche di write possono essere ordinate in modo da minimizzare il tempo di attesa
6. Ad esempio si possono selezionare le richieste da servire in base al minimo tempo di accesso, oppure
7. al minimo tempo di posizionamento stimato



Possono essere realizzati sia con hardware dedicato sia con software che usa hardware standard (esistono anche soluzioni ibride)

- *Soluzioni software* richiedono costi di CPU (cicli aggiuntivi)
- *Soluzioni hardware*
 - richiedono unità di controllo speciali che eseguono i calcoli di parità
 - hanno in genere velocità maggiore delle soluzioni sw. Dipende da:
 - dimensione della cache
 - quanto rapidamente i dati vengono scaricati sui dischi
 - supportano *hot swapping* (se possibile)



Data Striping

14

Striping

Aumenta le prestazioni

Distribuzione dei dati su multipli dischi

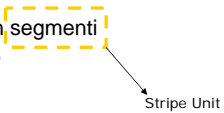
- In modo trasparente
- I dati sequenziali vengono suddivisi in segmenti
- Scritti con un algoritmo di *round robin*

Stripe Width

- Numero di dischi usati dallo striping
- Può non coincidere con il numero di dischi totali

Ridondanza

Aumenta l'affidabilità




Impianti Informatici

POLITECNICO DI MILANO

Il RAID si avvale di due tecniche antitetiche,

- La ridondanza, per aumentare l'affidabilità del sistema
- E lo striping dei dati, per aumentarne le prestazioni
- Mediante lo striping i dati vengono distribuiti,
- in modo trasparente all'utente, su più dischi visti come un unico disco veloce e di grande capacità
- In dettaglio, i dati che devono essere scritti sequenzialmente, ad esempio un file, vengono suddivisi in segmenti che vengono memorizzati su più dischi fisici
- con un algoritmo di round robin
- Si definisce unità di stripe il segmento che viene scritto su un solo disco; la sua dimensione dipende dall'implementazione, in genere tra i 2 e i 128 kbyte
- L'ampiezza dello stripe è
- il numero di dischi usato dall'algoritmo di striping,
- e non necessariamente coincide con il numero di dischi fisici dell'array



Data striping: prestazioni

15

Parallelismo

- Più richieste contemporanee servite in parallelo
 - Si riduce il *queueing time*
- Una singola richiesta di I/O per *multiple block* può essere servita in parallelo da più dischi
 - Aumenta il transfer rate



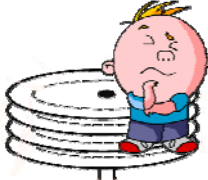
Impianti Informatici

POLITECNICO DI MILANO

Le prestazioni globali di I/O migliorano tramite Data Striping perché più operazioni di I/O

- vengono eseguite in parallelo, infatti:
- Più richieste contemporanee vengono eseguite in parallelo da dischi diversi,
- riducendo così il tempo di attesa di ogni richiesta davanti ai dischi.
- Inoltre una singola richiesta di I/O per multiple block può essere servita da più dischi che operano in modo coordinato,
- aumentandone il transfer rate

Vulnerabilità 16

$$P_{\text{guasto}}[100 \text{ dischi}] = \sim 100 * P_{\text{guasto}}[1 \text{ disco}]$$


↓

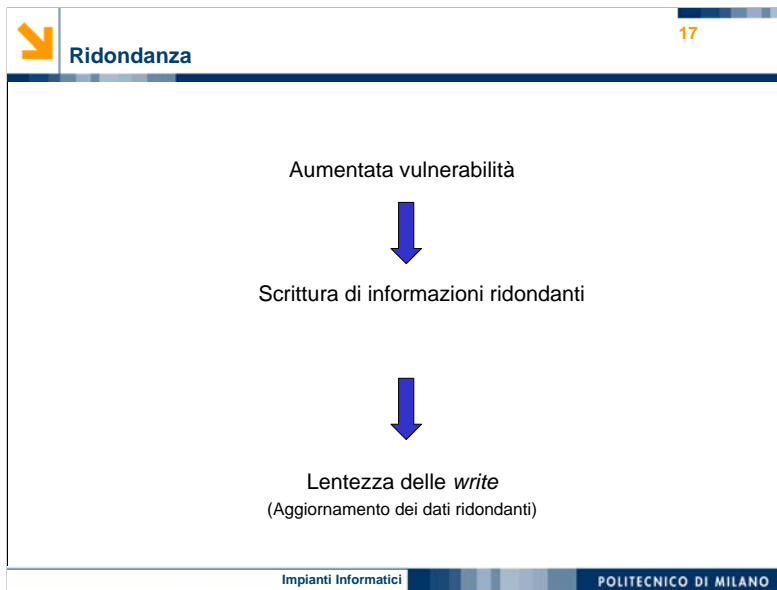
MTTF(1 disco) = 200000 = ~23 anni
 MTTF(100 dischi) = 2000 = ~3 mesi

Ridondanza → Dati ridondanti memorizzati su altri dischi

Correzione errori
 Recupero dati persi

Impianti Informatici POLITECNICO DI MILANO

1. L'uso di più dischi, porta ad un'aumentata vulnerabilità del sistema. Più elevato è il numero di dischi dell'array, **maggiori** sono i benefici prestazionali ma **minore** è l'affidabilità dell'intero array (cioè **maggiore** è la probabilità di guasti)
2. Infatti la probabilità di guastarsi di un array di 100 dischi, è 100 volte superiore a quella di un singolo disco
3. Così, se un disco ha un MTTF (Mean Time To Failure) di circa 23 anni,
4. un array di 100 dischi avrà un MTTF di circa 3 mesi
5. **La soluzione che si adotta per aumentare l'affidabilità di multipli dischi è la ridondanza** dei dati scritti: in questo modo si ha la possibilità
6. di correzione di eventuali errori o
7. perdite di dati in caso di disco guasto, con tecniche di codici a correzione di errore che utilizzano
8. informazioni ridondanti memorizzate su dischi diversi da quelli sui quali vengono scritti i dati



1. L'introduzione della ridondanza, per compensare l'aumentata
2. vulnerabilità dell'uso di più dischi, comporta un peggioramento
3. delle prestazioni in scrittura del RAID, perché le write devono aggiornare anche
4. le informazioni ridondanti, quindi sono più lente delle tipiche operazioni di scrittura

Unità di stripe 18

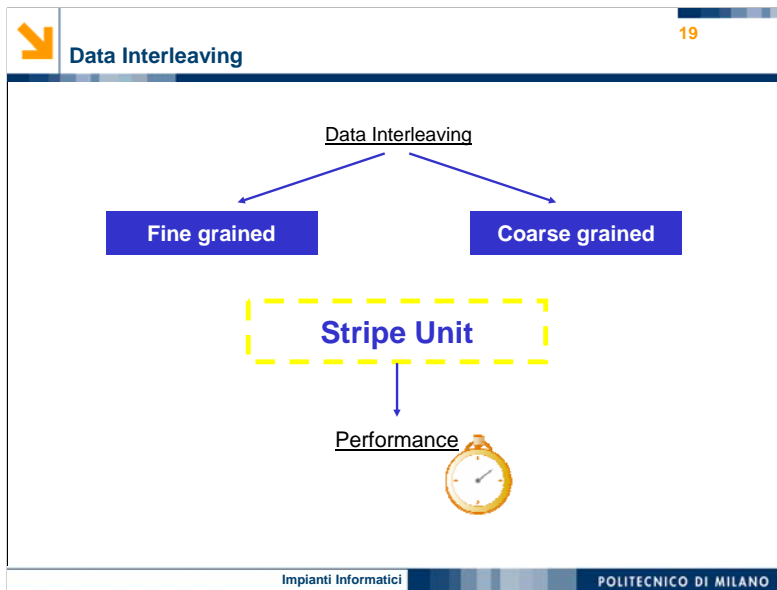
La dimensione influenza le prestazioni

Stripe unit piccola???

<u>SVANTAGGI</u>	<u>VANTAGGI</u>
Le singole richieste si distribuiscono su più dischi	Evita dischi con utilizzo asimmetrico (<i>access skew</i>)
Riduce l'efficacia di tecniche quali il <i>prefetching</i> dei dati	

Impianti Informatici POLITECNICO DI MILANO

1. La scelta dell'unità di stripe influenza in modo consistente le prestazioni del RAID:
2. Una piccola unità può dare luogo a singole richieste che
3. si distribuiscono su più dischi con un aumento del numero di operazioni di I/O e di dischi occupati. Alcune tecniche,
4. come ad esempio quella di prefetching, che vedremo in seguito, possono perdere di efficacia dato che i dati si trovano fisicamente dispersi e mescolati fra molteplici dischi
5. Il vantaggio di stripe unit ridotte è che in questo modo si diminuisce la probabilità di avere un sottoinsieme di dischi con un utilizzo sproporzionatamente asimmetrico.



1. Come accennato, le dimensioni dell'unità di stripe determinano in modo significativo
2. le performance dell'array di dischi.
3. A seconda delle dimensioni delle stripe unit si parla di Data interleaving
4. di tipo Fine grained
5. oppure coarse grained.

Data Interleaving: fine grained

20

Dati scomposti in stripe unit piccole
 Tutte le richieste di I/O possono usare l'intero array

VANTAGGI

Elevato transfer rate

SVANTAGGI

Serve una singola richiesta logica di I/O alla volta
 Attesa per il posizionamento di ciascun disco

Latenza rotazionale di 1 disco

Impianti Informatici
POLITECNICO DI MILANO

1. I dati vengono suddivisi in piccole unità in modo tale che tutte le richieste di I/O, indipendentemente dalle loro dimensioni,
2. possano utilizzare tutti i dischi dell'array.
3. Il vantaggio è un elevato transfer rate per tutte le richieste di I/O, a fronte di
4. poter servire una sola richiesta logica di I/O alla volta perché vengono usati molti dischi.
5. Potrebbero inoltre essere problemi di attesa dovuti al posizionamento di ciascun disco coinvolto.

 **Data Interleaving: coarse grained** 21

Dati scomposti in *stripe unit* grandi

- Richieste di I/O *piccole* → Usano pochi dischi
- Richieste di I/O *grandi* → Usano tutti i dischi

Molte richieste di I/O *piccole* servite in parallelo

- Vengono servite più richieste logiche contemporaneamente

Richieste di I/O *grandi* con elevato transfer rate

- Accesso contemporaneo a multipli dischi

Impianti Informatici POLITECNICO DI MILANO

1. I dati vengono suddivisi in unità relativamente grandi in modo tale che tutte le richieste di
2. I/O di piccole dimensioni necessitano soltanto di pochi dischi,
3. mentre grandi richieste di I/O possono utilizzare tutti i dischi dell'array
4. Il vantaggio è che molte piccole richieste di I/O possono essere eseguite in modo concorrente,
5. Mentre le Richieste di I/O particolarmente grandi possono avere elevato transfer rate sfruttando l'accesso a molti dischi contemporaneamente


➔
RAID: Prestazioni di un'operazione di I/O
22

Metriche

- Response Time
- Throughput
 - Può essere stimato (ottimisticamente) come il reciproco del *Service Time* (con $U=1$)

Benchmarking:

La misurazione delle prestazioni deve avvenire prima che la richiesta logica di I/O venga ricevuta dal *volume manager*, dove viene scomposta nelle richieste fisiche ai dischi



Impianti Informatici
POLITECNICO DI MILANO

1. Come visto durante l'analisi dei dischi magnetici, due metriche importanti per la misurazione della prestazioni di un'operazione di I/O sono
 2. il *response time*
 3. e il *throughput*. Il reciproco del *service time*
 4. è una stima ottimistica del *throughput massimo*
- Le prestazioni I/O possono essere misurate a diversi livelli della gerarchia di storage.
5. Per quantificare gli effetti delle diverse tecniche di ottimizzazione bisogna misurare i tempi da quando la richiesta è consegnata al sistema storage prima che venga potenzialmente spezzata dal "volume manager" in richieste dirette a dischi multipli.

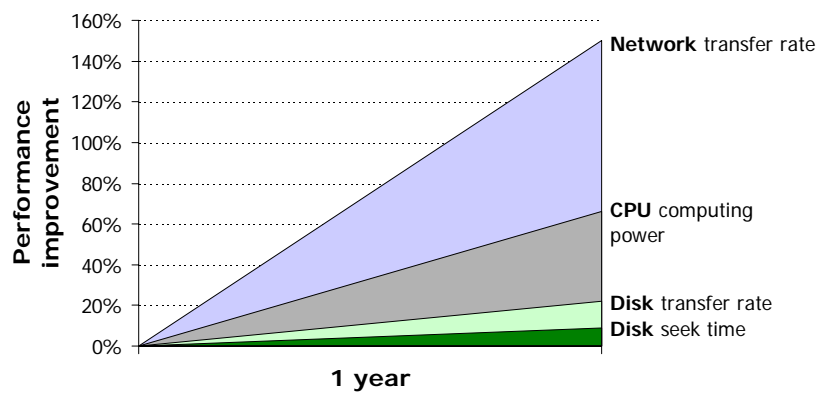


Evoluzione delle capacità (legge di Moore)

23

Legge empirica che all'origine riguardava l'andamento della densità dei transistori per chip (che raddoppia ogni 18 mesi)

concerne l'incremento della *capacità operativa* (che si moltiplica per 100 ogni 10 anni)



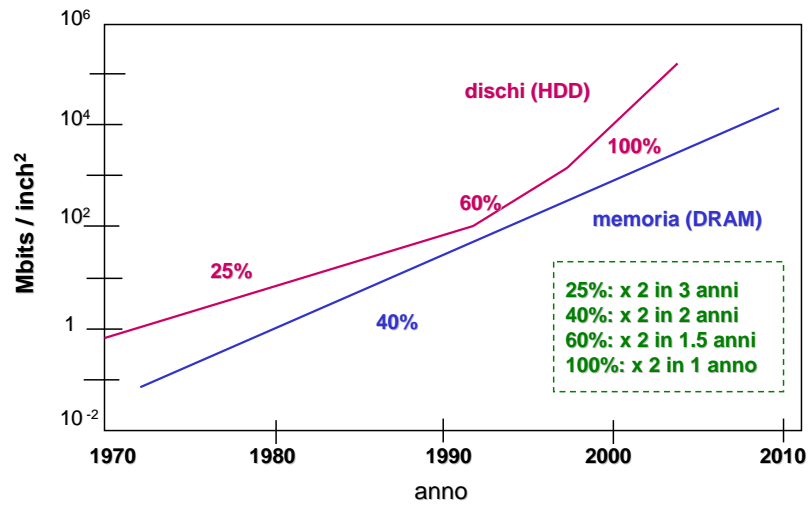
Impianti Informatici

POLITECNICO DI MILANO



Esempio: evoluzione dischi (densità superficiale)

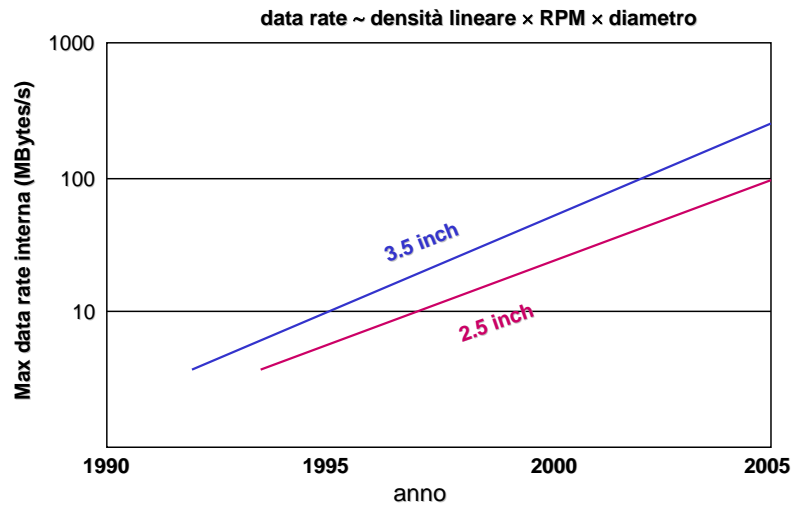
24





Esempio: evoluzione dischi (data rate)

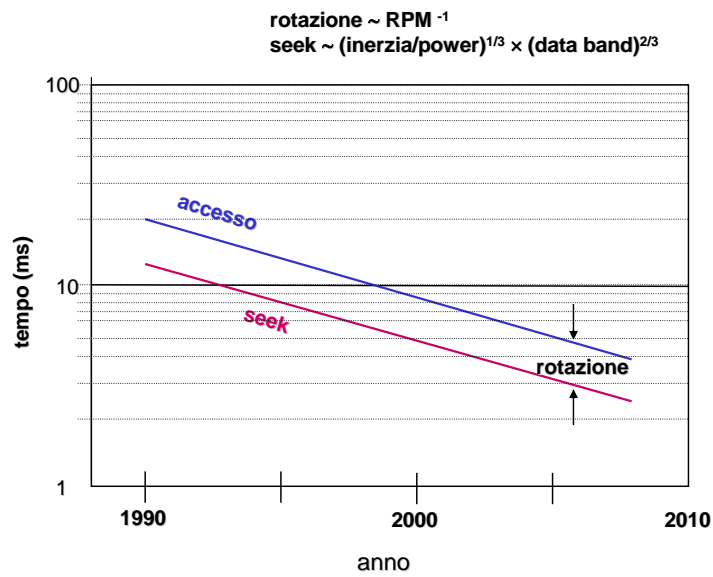
25





Esempio: evoluzione dischi (tempi di accesso)

26





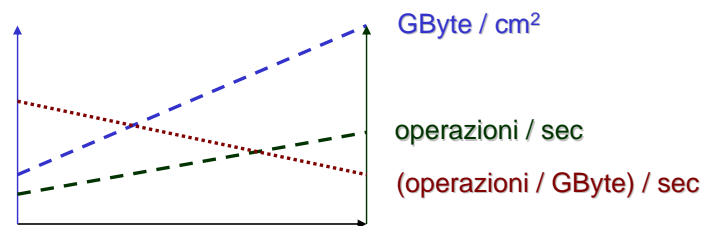
la crescita con **tassi diversi** di alcune grandezze può dare luogo ad alcuni problemi:

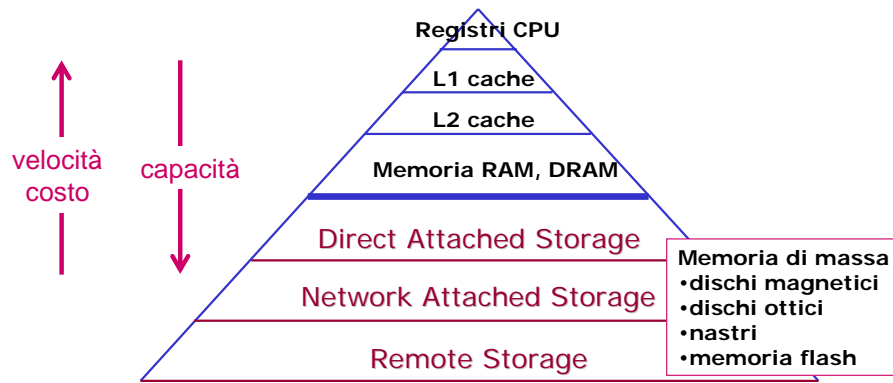
- in particolare la capacità di memoria cresce più rapidamente dei tempi di accesso perciò la densità degli accessi è diminuita nel tempo
- pericolo di non completo utilizzo (inedia o “starvation”) dei dispositivi (processori e dischi)

la capacità dei dischi è cresciuta dal 1956 di 5×10^7 volte,

entro 4 anni si pensa di raggiungere 500 Gbit per inch²

con metodi olografici 1 Tbit può essere contenuto in un volume di 1 cm³ (fonte: *Scientific American* aug. 05)





 **Ottimizzazione delle prestazioni** 29

Read Caching

Prefetching

Write Buffering

- Quando i dati sono suddivisi su più dischi (*striping*) la *locality* è modificata
 - le regioni attive possono essere contigue su più dischi così che i bracci di posizionamento delle testine hanno una estensione di movimento più limitata
 - lo stesso carico utilizza su ognuno dei diversi dischi solo una frazione dello spazio che userebbe su uno.

Impianti Informatici POLITECNICO DI MILANO

Esistono varie tecniche per migliorare le prestazioni di un sistema RAID; tra queste le più usate sono

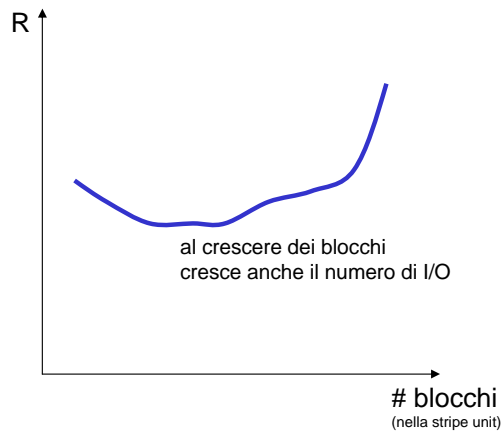
1. Il caching dei dati letti
2. Il prefetching delle informazioni
3. E l'uso i buffer per la scrittura



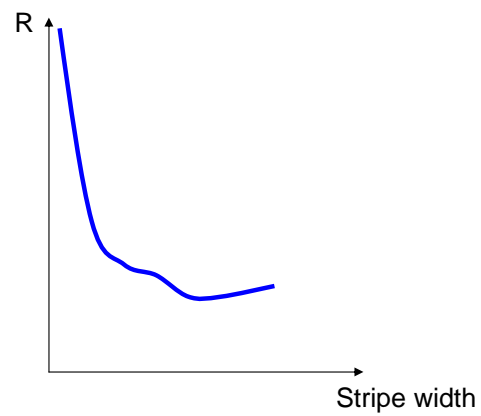
Locality in presenza di Striping

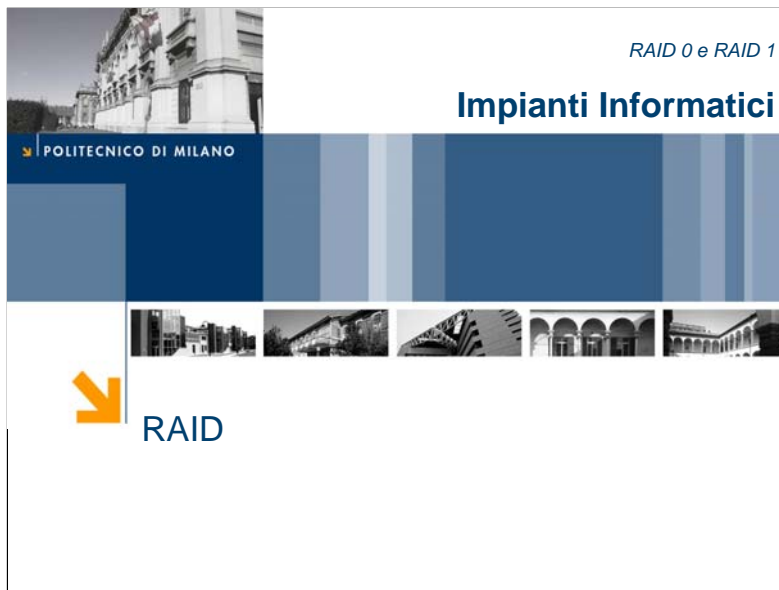
30

Resp. time in funzione della
dimensione della stripe unit
(numero di blocchi di una
stripe) parità
di no. dischi



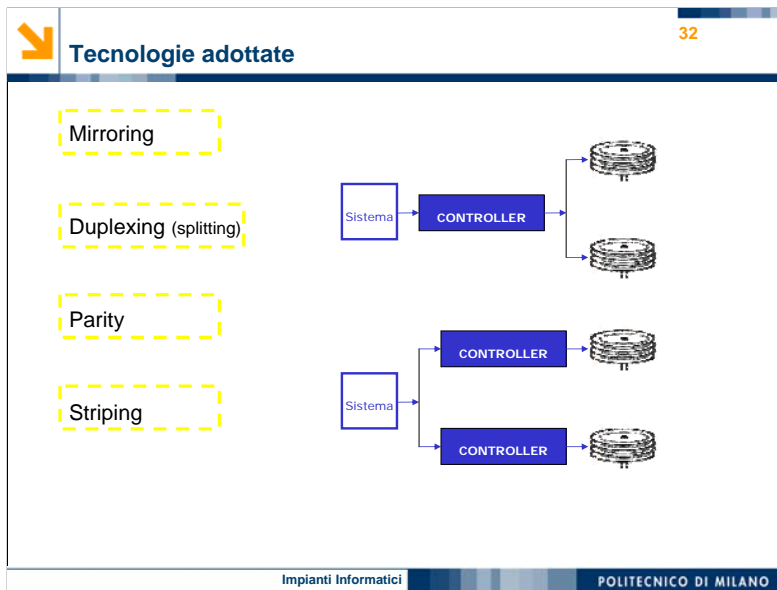
Resp. time in funzione
del numero di dischi
a parità di blocchi di stripe






Hot Word:

- Mirroring (slide 9)
- RAID 0
- RAID 1
- RAID 0+1
- RAID 1+0



Ci sono molteplici modi di implementare un array di dischi in RAID, usando combinazioni di diverse tecnologie,

- quali il mirroring,
- il duplexing,
- il parity
- e lo striping,
- Il primo si avvale di
- coppie di dischi, ciascuna contenente gli stessi identici dati replicati,
- garantendo in questo modo un'elevata affidabilità del sistema. Una sua estensione, ancor più sicura,
- è detta duplexing, in cui,
- oltre al disco, viene replicato
- anche l'hardware che lo controlla.
- Un'alternativa al mirroring è l'uso di informazioni di parità, che bilanciano affidabilità del sistema e sfruttamento ottimale delle risorse.
- Infine lo striping, come visto, consiste nello scomporre l'informazione da memorizzare in sottoparti, andando a utilizzare in parallelo molteplici dischi.



I livelli di RAID

33

Non esiste un unico tipo di RAID
Ci sono molti *livelli*

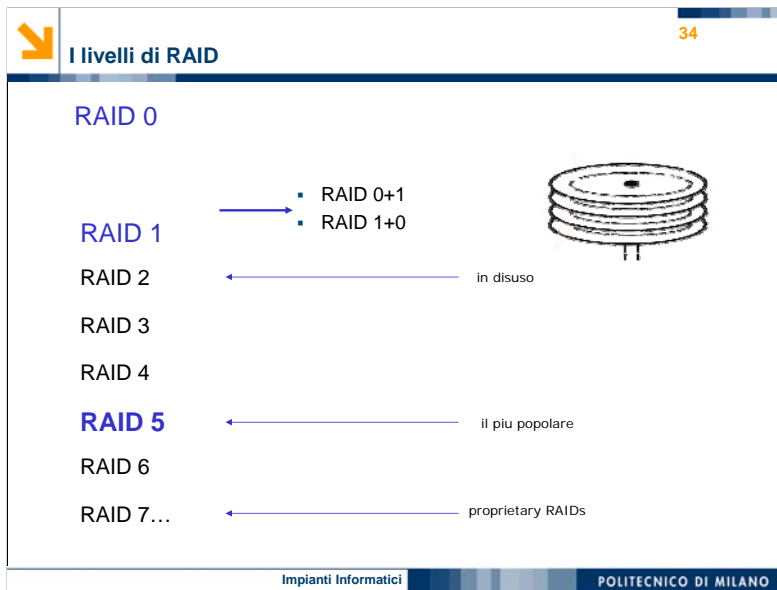
- Tecnologia
- Configurazione
- Obiettivi

Il *controller* determina quali livelli possono essere implementati

- Per alcuni livelli non è necessario un *controller RAID*
 - Sistema operativo
 - Software di management dell'array


Impianti Informatici POLITECNICO DI MILANO

1. Non esiste un unico tipo di RAID, ma ci sono
2. diversi livelli, con differenti
3. tecnologie usate, alternativi
4. modi di configurare lo stesso insieme di dischi,
5. per far fronte a diverse esigenze.
6. I livelli di RAID differiscono in termini del controller richiesto per essere implementati. In generale, semplici (ed economici) controller non implementano i livelli i RAID piu` avanzati e complessi.
7. Alcuni livelli non richiedono nemmeno un controller dedicato, e le loro funzionalita` possono essere svolte direttamente
8. a livello di sistema operativo o di altro
9. software per il management dell`array di dischi.



I semplici tra i livelli raid, spesso supportati direttamente a livello software e da controller anche di fascia bassa,

- sono il RAID 0,
- il RAID 1
- e le loro combinazioni 0+1 e 1+0.
- Diversi controller, anche non particolarmente costosi, implementano il più popolare dei livelli RAID, il 5.
- Esistono anche implementazioni software, con pesanti svantaggi in termini di prestazioni. Esistono altri livelli, anche se meno comuni,
- che sono il RAID 3,4,6 e 7,
- oltre al RAID 2, particolarmente complesso da richiedere hardware proprietario,
- e perciò sempre più in disuso.



Requisiti dei dischi

35

Numero *minimo* di dischi


- Tecnologie implementate
 - RAID 0 (striping): ≥ 2 dischi
 - RAID 1 (mirroring): $(\geq) = 2$ dischi
 - Striping+parity: > 3 dischi

Numero *massimo* di dischi

- Limitato dal controller

Funzionamento ideale con dischi

- Identici
- Stessa capacità

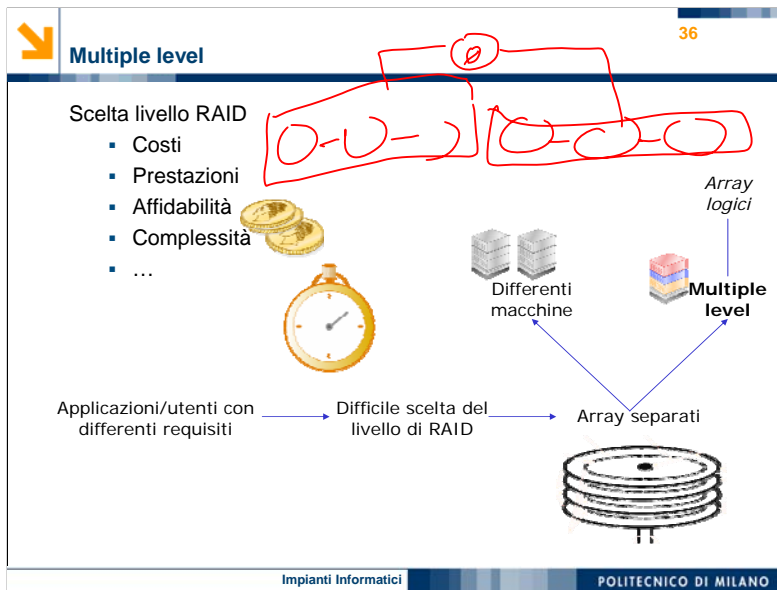


Impianti Informatici

POLITECNICO DI MILANO

Differenti livelli di RAID hanno diversi requisiti in termini di hard disk usati nell'array.

- La più importante differenza è legata al numero minimo di dischi necessari.
- Essa dipende dalle tecnologie implementate nel livello di RAID. Ad esempio,
- il semplice RAID0, che esegue solamente striping dei dati, richiede almeno due dischi,
- così come il RAID1 che si basa sul mirroring.
- Avere striping e parità richiede almeno 3 dispositivi, due per lo stripe e uno per la parità.
- Il massimo numero di dischi è invece limitato
- quasi esclusivamente dal controller, e non dal tipo di livello di RAID.
- Come ultima nota è bene evidenziare il fatto che indipendentemente dal livello di RAID, il funzionamento è ottimale quando si usano
- dischi identici
- di identica capacità.



1. La scelta di un livello di RAID è un compromesso tra diversi fattori,
2. come costi,
3. prestazioni,
4. affidabilità,
5. complessità..
6. Se si ha un insieme di applicazioni o utenti, con diversi requisiti,
7. può non essere immediato trovare il livello di RAID più adatto.
8. In questo caso, potrebbe essere conveniente creare due array separati, usando differenti livelli di RAID per ciascuno.
9. Una via semplice è quella di usare differenti macchine in cui implementare diversi livelli.
10. Alternativamente molti controller offrono la possibilità di configurare livelli multipli all'interno della stessa macchina,
11. usando quindi un unico array fisico e scomponendolo in array logici.

RAID 0: striping 37

Striping
No ridondanza

Redundant Arrays of Independent Disks

Dati


- Suddivisi in blocchi sequenziali
- Algoritmo di *striping* per distribuirli tra i dischi fisici




Numero minimo di dischi: 2


Impianti Informatici POLITECNICO DI MILANO

L'unica tecnica usata dal RAID 0

- è lo striping dei dati, senza
- l'introduzione di alcun tipo di ridondanza, tanto che alcuni non lo considerano un vero e proprio RAID.
- i dati scritti su un disco logico sono suddivisi
- in *blocchi* sequenziali
- e distribuiti tra i dischi fisici con un algoritmo di striping.
- richiede almeno 2 drive per poter essere implementato.


 **RAID 0: striping** 38

<p> <u>Vantaggi</u></p> <p>Costo minimo di implementazione </p> <ul style="list-style-type: none"> ▪ Massima capacità ▪ No ridondanza <p>Elevate prestazioni</p> <ul style="list-style-type: none"> ▪ Parallelismo di dischi e canali <p>Write efficienti</p> <ul style="list-style-type: none"> ▪ Non c'è ridondanza da aggiornare 	<p> <u>Svantaggi</u></p> <p>No dischi <i>hot spares</i></p> <p>Bassa affidabilità</p> <ul style="list-style-type: none"> ▪ No fault-tolerance ▪ No correzione errori
---	---



Impianti Informatici POLITECNICO DI MILANO

1. I vantaggi di questo livello sono
2. il costo minimo, grazie al
3. massimo sfruttamento delle capacità dello storage,
4. dato che non c'è alcuna ridondanza.
5. Ha inoltre elevate prestazioni, perché si avvale del
6. parallelismo di dischi e canali.
7. Anche le operazioni di scrittura sono particolarmente efficienti
8. perché non è necessario aggiornare alcuna forma di dati ridondanti.
9. Gli svantaggi sono il fatto che
10. non si possono usare dischi *hot spares*, una tecnica che consiste nel montare un numero superiore di drive nell'array, lasciandone alcuni in stand-by, e attivarli solo in caso di guasto.
11. Un altro evidente svantaggio del RAID 0 è la bassa affidabilità,
12. senza possibilità né di fault tolerance
13. né di correzione degli errori. È così raccomandato per dati non critici, e necessità di elevate prestazioni.



RAID 1: mirroring

39

Tutti i dati sono *duplicati* su un altro disco

- *Mirroring* (replica del disco)
- *Duplexing* (replica di disco e controller)

Numero minimo di drive: 2

<u>Vantaggi</u>	<u>Svantaggi</u>
<p>Elevata affidabilità</p> <ul style="list-style-type: none"> ▪ Fault-tolerance <p>Read efficienti</p> <ul style="list-style-type: none"> ▪ Tempo minimo tra i due drive ▪ Letture parallele (se un device è occupato si usa l'altro) 	<p>Costo</p> <p>Sfruttamento del 50% della capacità fisica</p>

Impianti Informatici
POLITECNICO DI MILANO

Con il RAID di livello 1

- tutti i dati vengono duplicati su un altro disco.
- È generalmente implementato mediante mirroring, anche se esistono sistemi
- con duplexing, ovvero replicazione sia del disco che del controller.
- Sono necessari almeno due drive per funzionare.
- La principale caratteristica è l'elevata
- affidabilità, garantendo sistemi altamente
- fault tolerant perché, qualora un disco si guastasse, è subito accessibile il suo gemello.
- Inoltre i tempi di lettura giovano particolarmente di questa configurazione,
- dato che il tempo di un'operazione di read sarà quello di quella più rapida tra i due dischi.
- La maggiore penalità è
- il costo, dovuto al fatto che viene sfruttata solamente
- il 50% della capacità di storage potenziale.

RAID 0+1
40

Tecnologie applicate ai dati:

- 1) Striping
- 2) Mirroring

High data transfer performance

Buona affidabilità

- Il guasto di un disco porta alla situazione RAID 0

Overhead elevato

stripe 1

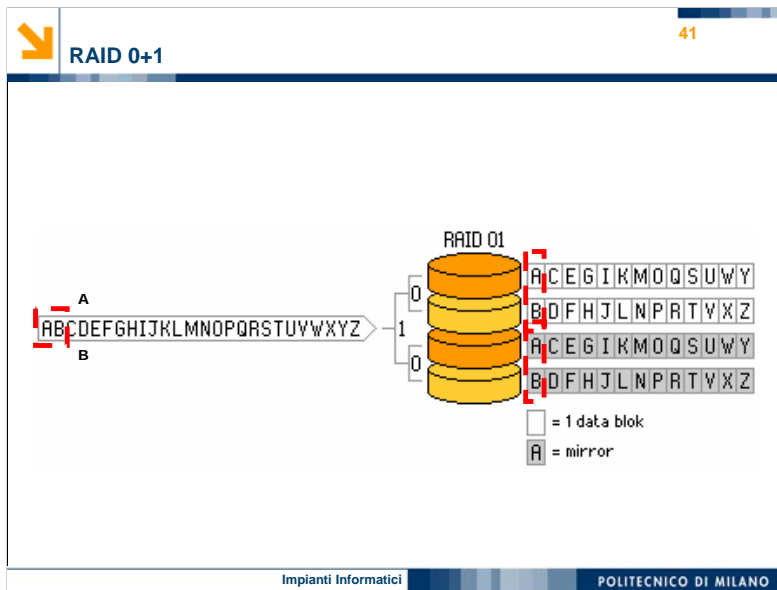
stripe 1

dati duplicati

Impianti Informatici
POLITECNICO DI MILANO

Vediamo ora due differenti modelli che combinano assieme i primi due livelli di raid, cercando di trarne i rispettivi vantaggi.

1. Il primo che analizziamo è il RAID 0+1.
2. Prima viene effettuato lo striping dei dati, secondo il livello 0,
3. Poi viene eseguito il mirroring degli stessi, secondo il livello 1.
4. Si ottiene così un sistema particolarmente performante,
5. nonostante la buona affidabilità.
6. Il guasto a un singolo disco porta nella situazione RAID 0
7. Così come nel RAID 1, la duplicazione dei dischi porta ad un elevato overhead causato dal sottosfruttamento della capacità.



1. La figura mostra con un esempio il funzionamento del RAID 0+1
2. I dati da memorizzare
3. Vengono suddivisi, secondo l'algoritmo di striping, tra i dischi
4. Contemporaneamente vengono duplicati nei dischi replica.

RAID 1+0 42

Tecnologie applicate ai dati:
 1) Mirroring
 2) Striping

Elevate prestazioni
 Fault-tolerance

Numero minimo di dischi: 4

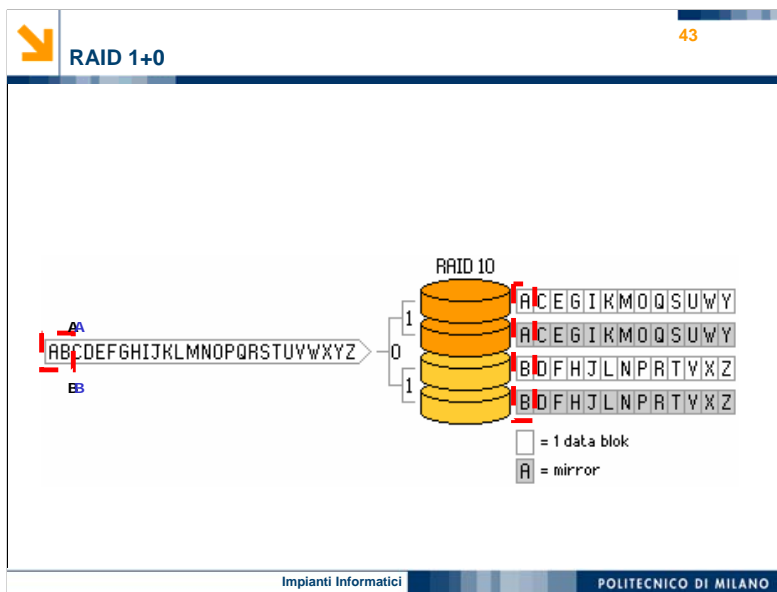
Costo
 ▪ Sfrutta solo 50% capacità fisica

diagramma: stripe 1, disk array - 4 dischi, dati duplicati

The diagram illustrates the RAID 1+0 configuration. It shows a 'stripe 1' of data being mirrored onto two pairs of disks in a 'disk array - 4 dischi'. The data is first mirrored (A, B) and then striped across the two pairs. Handwritten red annotations show the data flow and the 50% capacity utilization.

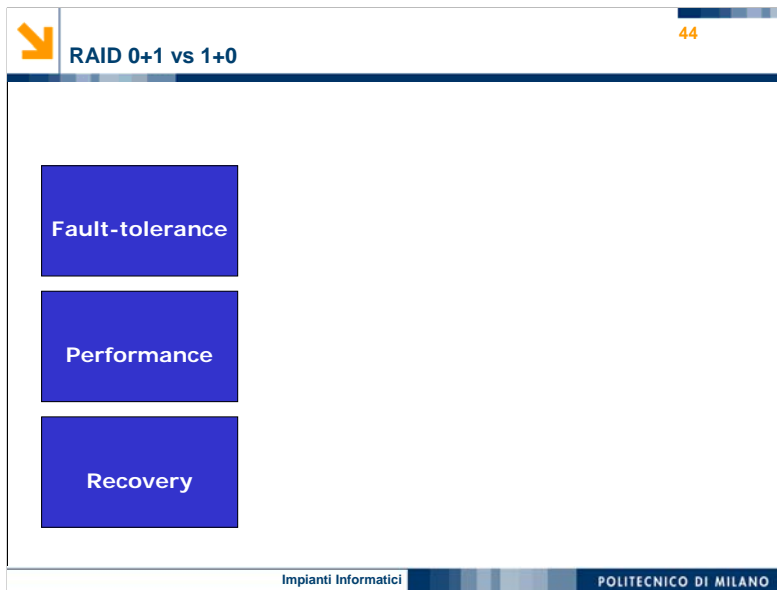
Il RAID 1+0 funziona similmente al RAID 0+1; vengono semplicemente invertite le operazioni.

1. Perciò viene prima eseguito il mirroring delle informazioni,
2. Quindi lo striping.
3. Si ottiene così un'architettura che unisce elevate prestazioni
4. ed elevata tolleranza ai guasti.
5. Richiede un minimo di 4 dischi, e come il precedente livello
6. È penalizzato da un forte costo per il basso sfruttamento dei dischi.



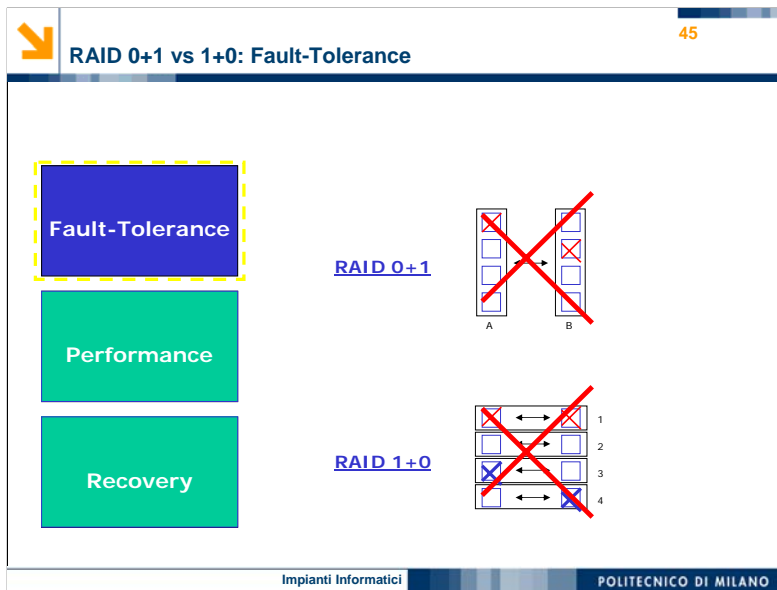
Lo schema in figura mostra come le informazioni

1. vengano memorizzate sui diversi dischi,
2. Cioè la scrittura contemporanea sui dischi gemelli,
3. Di sottoparti del file generate dall'algoritmo di striping

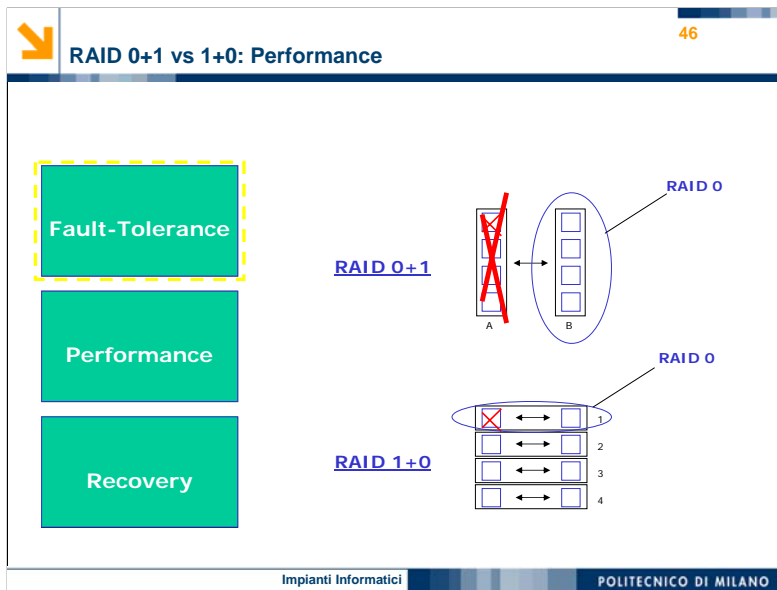


Possiamo analizzare le differenze tra i due livelli di RAID appena illustrati per quanto riguarda:

1. La tolleranza ai guasti
2. Le prestazioni in caso di guasto ad un disco ed
3. il relativo tempo di ripristino del sistema una volta che il disco è stato ripristinato

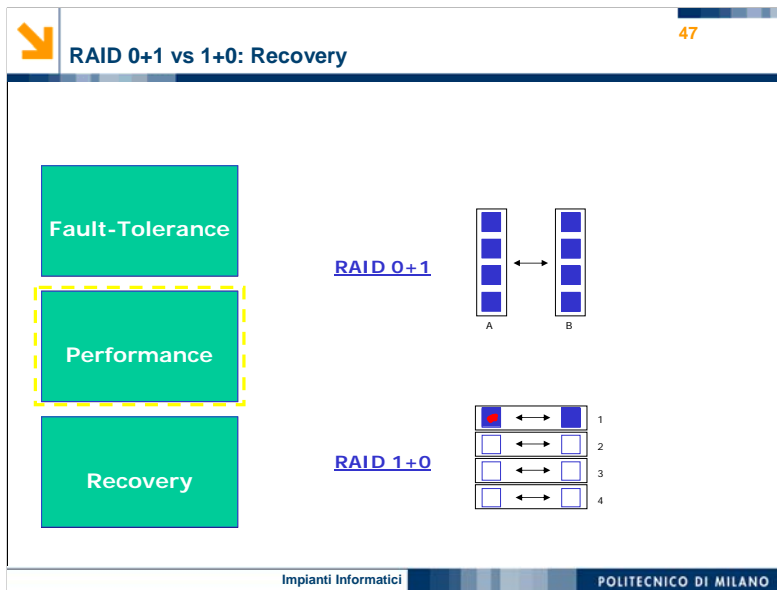


1. Nel RAID 0+1, e' sufficiente perdere un drive da ciascun set di dischi per avere il fallimento dell'intero sistema,
2. Cioe' sia un disco dall'insieme A, sia da quello B. Infatti, dato che si esegue prima lo stripe, i dati da memorizzare sono suddivisi tra molteplici dischi.
3. NEI RAID 1+0, invece, occorre perdere tutti i dischi appartenenti allo stesso mirror;
4. ad esempio, nel caso in figura, entrambi i dischi dell'insieme 1,
5. Mentre perdere due dischi non accoppiati non comporta la caduta del sistema complessivo.



Mentre le prestazioni in condizioni di funzionamento normale sono equivalenti,

- in caso di danneggiamento di un disco i due livelli di RAID si comportano in maniere leggermente differente.
- Nel RAID 0+1, la perdita di un disco
- porta al fallimento dell'intero insieme, quindi il sistema continuerà a funzionare
- come se fosse in RAID 0.
- Nel RAID 1+0, la perdita di un disco, ha l'effetto di far operare in modalità
- RAID 0 solo l'insieme coinvolto, mentre il resto del sistema funziona normalmente



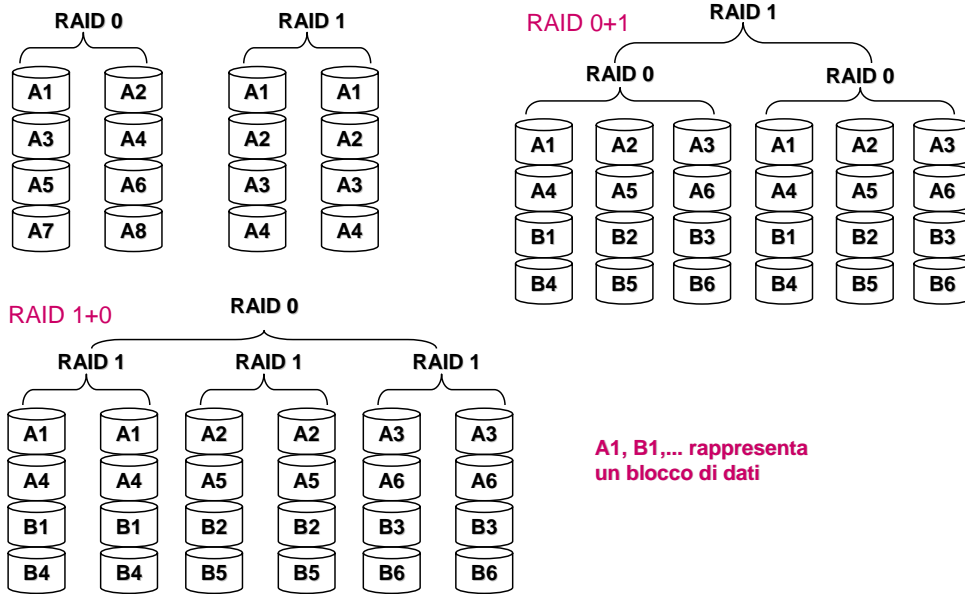
Un'ultima differenza tra i livelli 1+0 e 0+1 riguarda la velocità di ripristino del sistema quando, dopo il guasto di un disco, ne viene effettuato il ripristino.

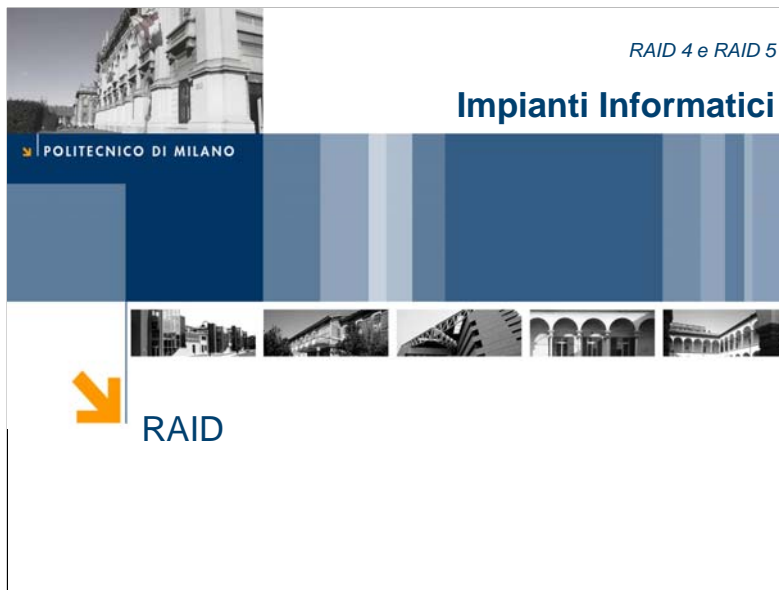
- Il RAID1+0 ha un solo disco da ripristinare,
- facendo il mirror del disco rimasto,
- Il RAID0+1 deve invece effettuare il mirror
- di tutto l'insieme, ed è quindi più lento da eseguire



schema RAID 0, 1, 0+1, 1+0

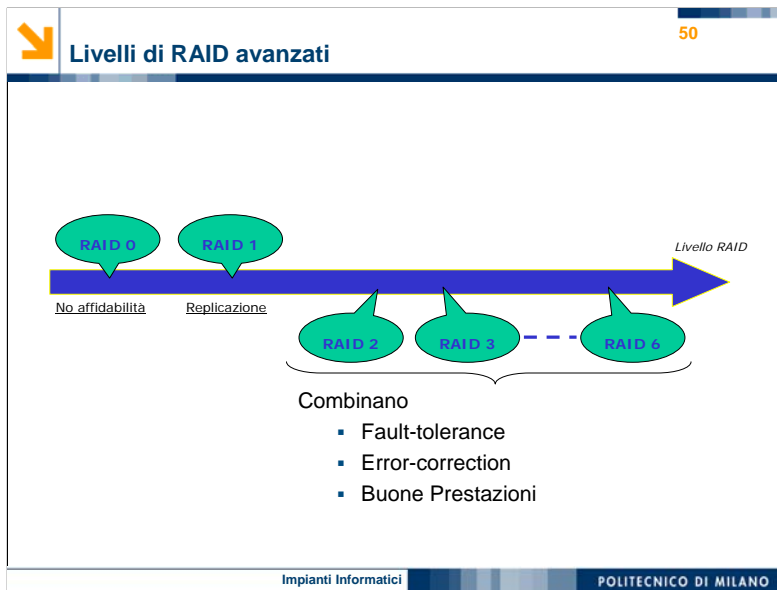
48





Hot Word:

- Mirroring (slide 9)
- RAID 0
- RAID 1
- RAID 0+1
- RAID 1+0

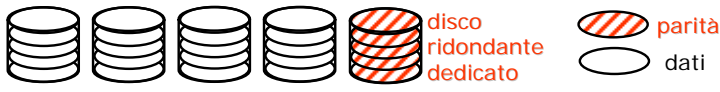


1. Mentre i livelli 0 e 1, e le loro combinazioni,
2. o non implementano alcuna forma di affidabilità,
3. o la implementano in maniera piuttosto grossolana, ovvero duplicando grezzamente le informazioni, con ingente spreco di capacità fisica,
4. i livelli superiori hanno funzionalità di
5. fault-tolerance e
6. error-correction più avanzate,
7. pur rimanendo efficienti dal punto di vista prestazionale

➔ RAID 4: block interleaved parity
51

Unità elementare: *blocco*

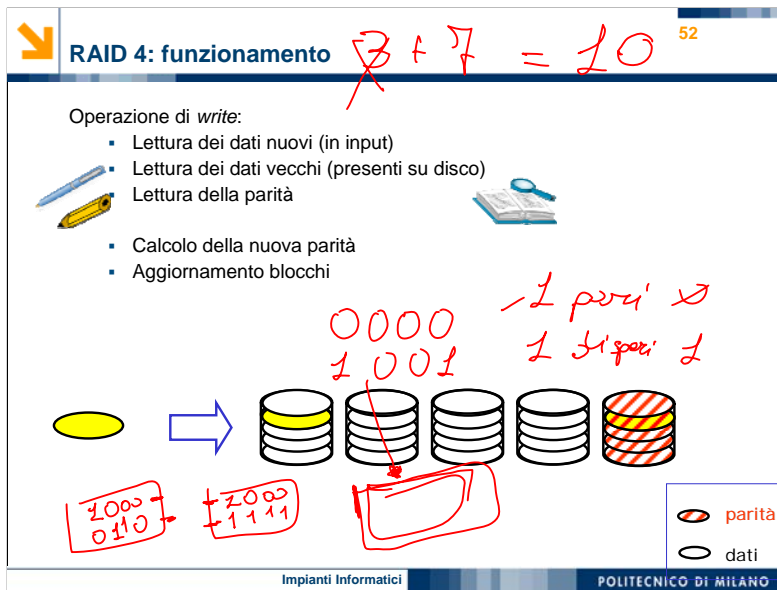
- read inferiori ad un blocco usano un solo disco



Impianti Informatici
POLITECNICO DI MILANO

Nel RAID di livello 4,

- l'unità elementare dei dati è il blocco. Essi sono perciò distribuiti tra i dischi in blocchi, e non in bit
Le operazioni di I/O in lettura inferiori ad un singolo blocco
- utilizzano un unico disco, mentre quelle che coinvolgono multipli blocchi,
- vengono suddivise tra più dischi.
- Il RAID 4 dedica un intero disco alla ridondanza, mediante il meccanismo
- della parità, che viene applicato a livello di blocco.



1. Quando si deve eseguire un'operazione di I/O in scrittura,
2. le write devono innanzitutto
3. leggere i nuovi dati da scrivere,
4. Ma anche quelli vecchi
5. e la parità,
6. Quindi si calcola il nuovo blocco di parità e
7. oltre a memorizzare i dati,
8. si aggiorna tale blocco

➔
Ciclo *Read-modify-write*
53

Sistema **read-modify-write**

- per ogni operazione di scrittura breve
- 4 accessi
 - due per leggere i dati vecchi e la parità vecchia
 - uno per scrivere i dati nuovi
 - uno per scrivere la parità nuova


Hand-drawn diagram illustrating the Read-Modify-Write cycle. It shows four boxes containing the numbers 3, 4, 5, and 1, followed by a plus sign and the number 13. Below this, a circular flow of four green arrows connects the steps: 'read' (from the data to the parity), 'modify' (calculating the new parity), 'write' (writing the new parity), and another 'read' (reading the new parity).

Impianti Informatici
POLITECNICO DI MILANO

Il meccanismo appena descritto per la scrittura di blocchi singoli, viene denominato

- Sistema Read-modify-write.
- Esso riguarda operazioni di scrittura che coinvolgono un solo disco, cioè di dati di dimensione minore ad una singola unità di stripe.
- per ogni operazione di scrittura di questo tipo, sono necessari quattro accessi al disco:
- due per leggere i dati vecchi e la parità vecchia
- allo scopo di calcolare la parità nuova
- uno per scrivere i dati nuovi
- uno per scrivere la parità nuova

RAID 4: caratteristiche
54



Affidabilità

- RAID 4 guasto con due dischi guasti


Disco ridondante

- Acceduto da ogni write
- Possibile bottleneck

È possibile usare dischi *hot spares*



Impianti Informatici
POLITECNICO DI MILANO

1. Nel RAID 4 viene dedicato un unico disco per la ridondanza.
2. Tale supporto deve essere acceduto
3. ad ogni operazione di scrittura, in modo da mantenere aggiornate le operazioni di parità,
4. Cosicché diventa facilmente il bottleneck del sistema.
5. Dal punto di vista dell'affidabilità,
6. il livello di RAID 4 si guasta se ci sono due dischi guasti contemporaneamente.
7. Tra le varie caratteristiche vi è la possibilità di utilizzare i dischi hot spares

 RAID 4: prestazioni 55

Lettura

- Veloce
- Parallelismo

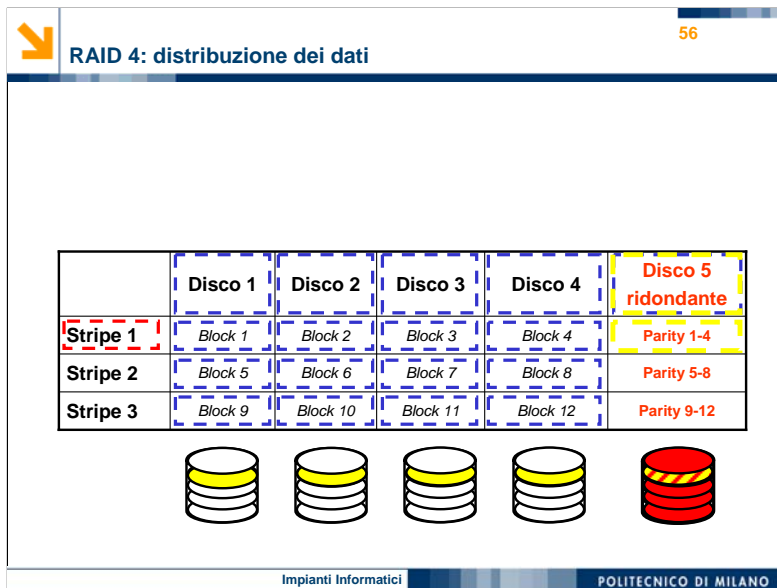



Scrittura

- Lenta
- Penalizzata dal *parity block*

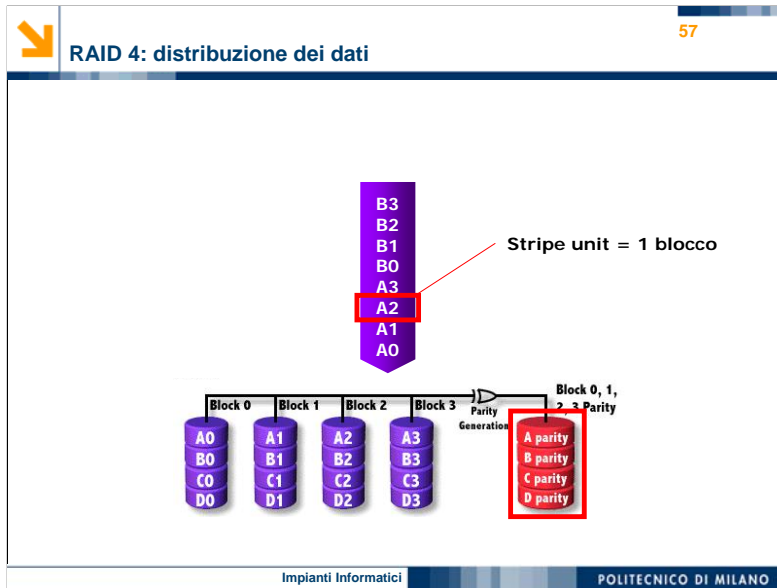
Impianti Informatici POLITECNICO DI MILANO

1. Analizzandolo dal punto di vista prestazionale,
2. Il RAID 4 è caratterizzato da un'alta velocità di trasferimento in lettura,
3. Ma da una ridotta velocità in scrittura.
4. Infatti durante una lettura il sistema può sfruttare il parallelismo dei dischi,
5. Mentre in scrittura le operazioni relative al calcolo e
6. alla scrittura del blocco di parità ne rallentano l'esecuzione



La tabella presenta come vengono distribuiti i dati tra i vari dischi di un array RAID 4.

1. Supponendo di avere 5 dischi,
2. 1 viene dedicato esclusivamente per le informazioni di parità
3. I dati vengono quindi distribuiti usando come unità elementare il blocco.
4. Ad ogni stripe, corrispondente ad un insieme di blocchi tra molteplici dischi,
5. è collegato un blocco di parità relativo, che viene salvato nel disco ridondante



Graficamente, la distribuzione dei nel RAID 4 è quella rappresentata in figura,

1. Con i dati distribuiti tra i dischi a seconda della dimensione dell'unità di striping stabilita
2. E la parità accentrata in un unico disco

➔
RAID 5: block interleaved distributed parity
58

Soluzione ampiamente adottata
Versatile

- Prestazioni/affidabilità
- Costo minimo per la ridondanza

Blocchi di parità distribuiti su tutti i dischi fisici

RAID 4

RAID 5

parità

dati


Impianti Informatici
POLITECNICO DI MILANO

1. Il RAID 5 è la soluzione più adottata e versatile:
2. unisce massimi vantaggi in termini di prestazioni e affidabilità,
3. a minimi costi, riguardanti un limitato sottosfruttamento della capacità,
4. Il suo funzionamento è simile al RAID4; la differenza sta nel fatto che non viene dedicato un intero disco alla parità, che costituiva il principale difetto del RAID4,
5. In questo livello, invece, i blocchi di parità sono
6. distribuiti uniformemente su tutti i dischi fisici, evitando il bottleneck di avere la ridondanza concentrata in un unico supporto

➔
RAID 5: prestazioni
59


Write

- Più lente di RAID 0 e RAID 1
- Occorre scrivere su tutti i dischi




Read

- Più veloci di RAID 1
- Parallelismo



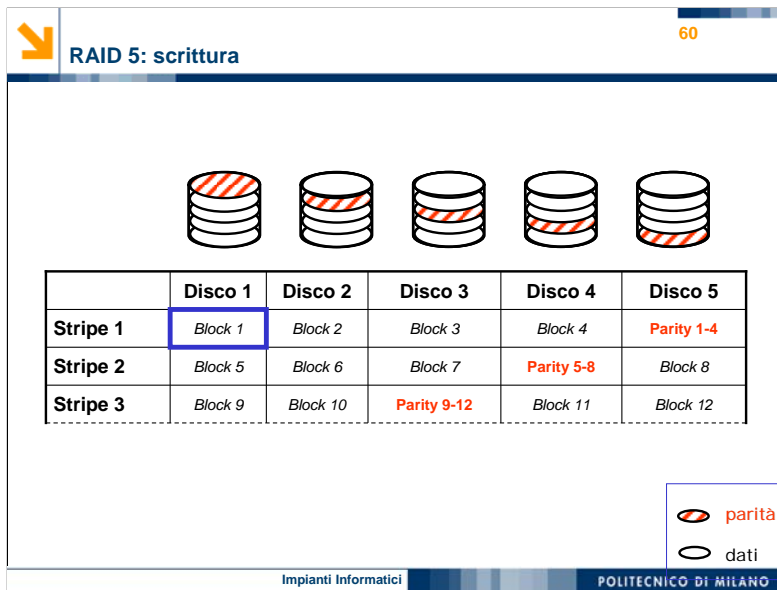
- In genere i blocchi di parità non sono acceduti nell'operazione di lettura.
- Vengono letti se un settore dà luogo a un errore *CRC* (Cyclic Redundancy Check): il settore errato viene ricostruito utilizzando le informazioni dei rimanenti blocchi della stripe unit in questione e del blocco di parità

Load balancing su tutti i dischi

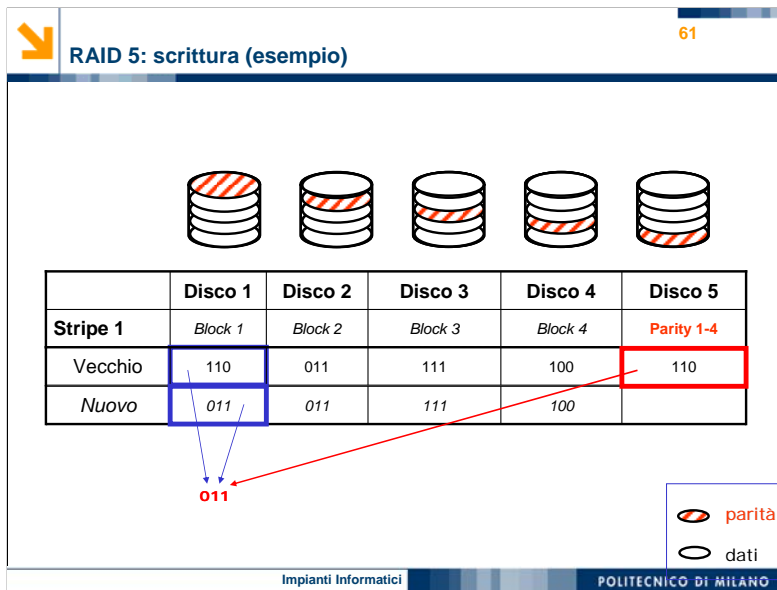


Impianti Informatici
POLITECNICO DI MILANO

1. Dal punto di vista delle prestazioni, i sistemi RAID 5 presentano
2. Operazioni di scrittura
3. più lente rispetto alle più semplici versioni di raid, la 0 e la 1, in quanto occorre effettuare le write
4. su tutti i dischi, e quindi attendere che abbiano finito tutti i drive
5. Relativamente alle operazioni di lettura il RAID 5 risulta
6. più veloce del RAID1, perché sfrutta in maniera migliore
7. il parallelismo del meccanismo di striping.
8. Come ultima considerazione il livello 5 effettua un efficiente load balancing su tutti i dischi dell'array.



1. Supponiamo ora di disporre di un sistema con 5 dischi fisici,
2. configurati come un disco logico RAID5
3. La distribuzione dei blocchi dati e di parità tra i dischi
4. avrà la struttura mostrata in tabella
5. L'esecuzione di un'operazione di scrittura, ad esempio sul blocco 1, richiede una serie di operazioni, corrispondenti al ciclo read-modify-write già analizzato



L'operazione di scrittura del blocco 1, comporta innanzitutto la lettura

- Del vecchio valore del blocco,
- E del valore di parità.
- Una volta ottenuti questi dati,
- assieme al valore del nuovo blocco da scrivere
- si può computare la parità aggiornata
- Che viene memorizzata su disco

Calcolo della parità
62

<u>A</u>		<u>B</u>		<u>C</u>		<u>D</u>		<u>Parità</u>
1	+	2	+	3	+	4	=	10

$1 + 2 + 4 = 7$

Impianti Informatici
POLITECNICO DI MILANO

Vediamo con un esempio semplificato come,

1. tramite le informazioni i parità, sia possibile ricostruire i dati originari in caso di guasto.
2. Supponiamo di avere unità di stripe contenenti numeri interi.
3. una possibile informazione di parità potrebbe essere la somma dei numeri stessi,
4. in questo caso pari a 10
5. Se ad esempio una informazione viene a mancare, per un qualsiasi guasto o errore
6. si può recuperare il dato a partire dalla parità e dai rimanenti blocchi.
7. Infatti sapendo che la somma deve essere 10
8. E che la somma parziale è pari a 7,
9. Il dato mancante non può che essere pari a 3.



Esempio di ridondanza

63

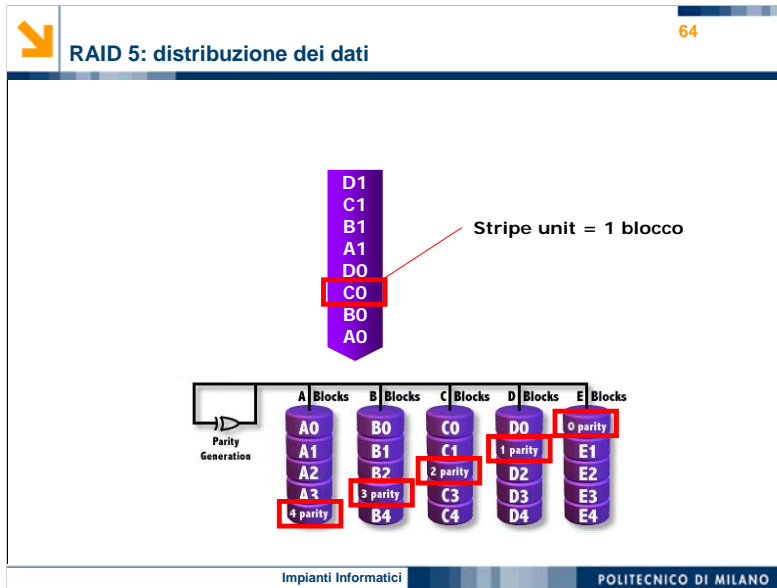
Disco 1	Disco 2	Disco 3	dati ridondanti
10	8	2	20
10	guasto	2	20

$$\text{guasto} = 20 - (10 + 2) = 8$$

Disco 1	Disco 2	Disco 3	parità
1	1	0	0
1	guasto	0	1

$$\text{parità} = \text{somma modulo } 2$$

$$\text{guasto} = \text{parità} - (1 + 0) = 0$$



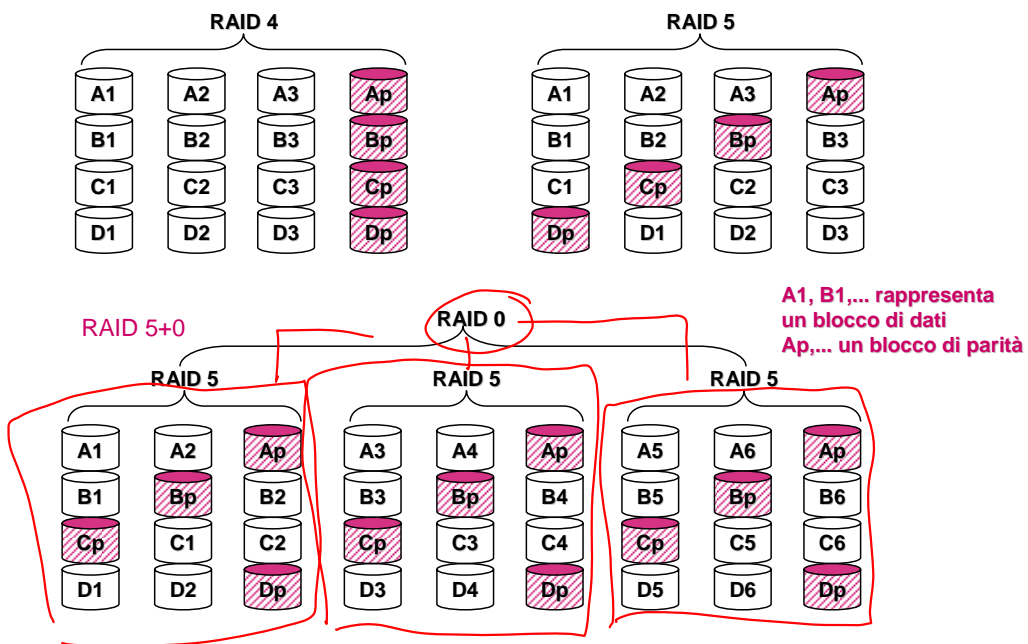
Graficamente, la distribuzione dei nel RAID 5 è quella rappresentata in figura,

1. Con i dati distribuiti tra i dischi a seconda della dimensione dell'unità di striping stabilita
2. E i blocchi di parità equamente ripartiti tra tutti i supporti



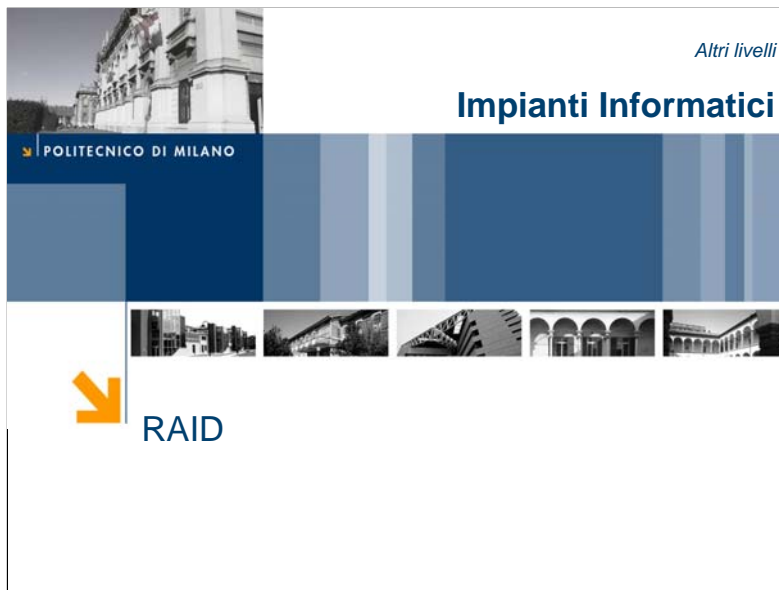
Schema RAID 4, 5, 5+0

65



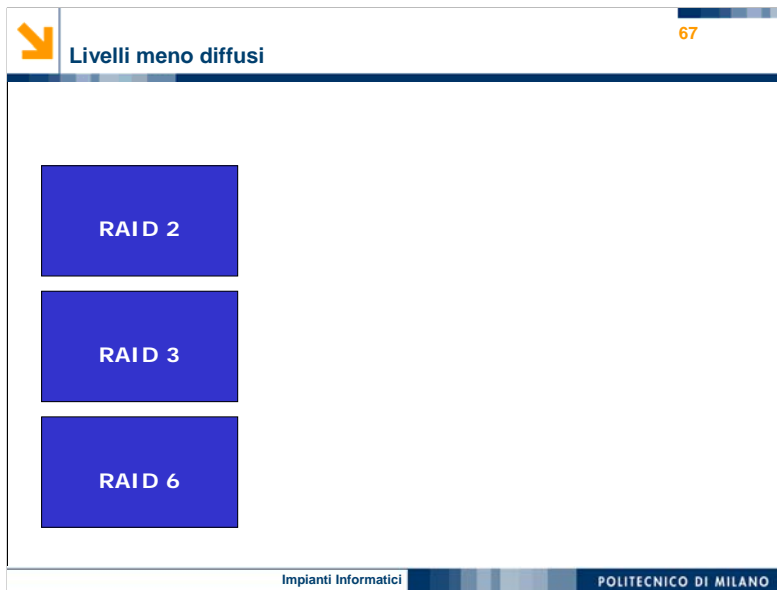
Impianti Informatici

POLITECNICO DI MILANO



Hot Word:

- Mirroring (slide 9)
- RAID 0
- RAID 1
- RAID 0+1
- RAID 1+0



Oltre ai semplici RAID 0 e 1, e ai RAID di livello più avanzato 4 e 5, esistono altri livelli, seppur meno diffusi dei precedenti.

In particolare si andranno ad analizzare,

- Il livello 2, che si avvale di codici di hamming per la ridondanza
- Il livello 3, molto simile al RAID4,
- E il livello 6, detto anche P+Q, che implementa una doppia ridondanza

RAID 2
68

Striping

- Dati divisi a livello di *bit*

Ridondanza

- Codici di *Hamming*
- In lettura viene verificata la correttezza dei dati e corretti gli errori su un singolo drive
- Individua 2-bit errors e corregge 1-bit errors on the fly

Affidabilità

Capienza

- con 8 dischi è 5 volte più capiente

Velocità (teorica)

- con 8 dischi è 5 volte più veloce
- non per accessi "piccoli"

RAID 2

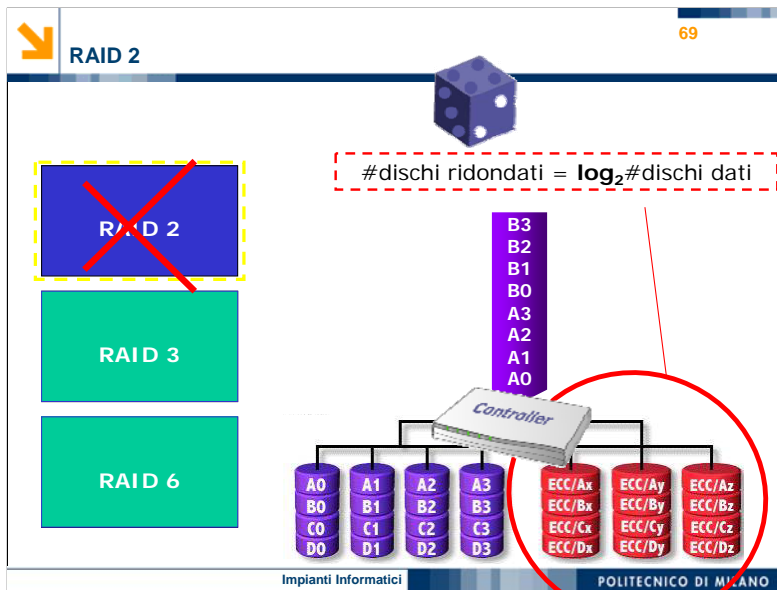
RAID 3

RAID 6

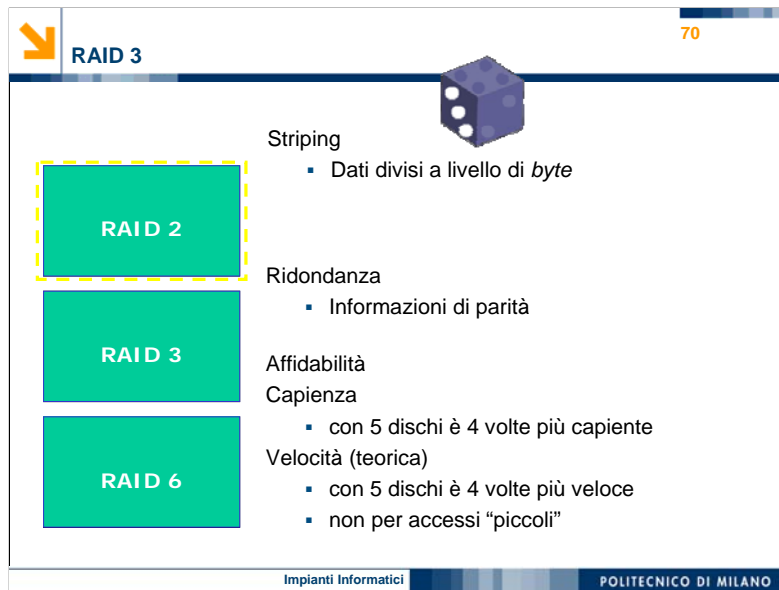
Impianti Informatici
POLITECNICO DI MILANO

Un sistema **RAID 2** implementa sia tecniche di

1. Striping
2. Che di ridondanza.
3. Esso divide i dati al livello di bit (invece che, come visto con altri RAID, di blocco) e usa
4. un codice di Hamming per la correzione d'errore.
Il RAID 2 presenta una
5. Buona affidabilità, grazie alla memorizzazione di informazioni ridondanti,
6. Aumentata capienza e
7. Aumentata velocità del sistema,
8. Soprattutto per accessi a file di grande dimensione.

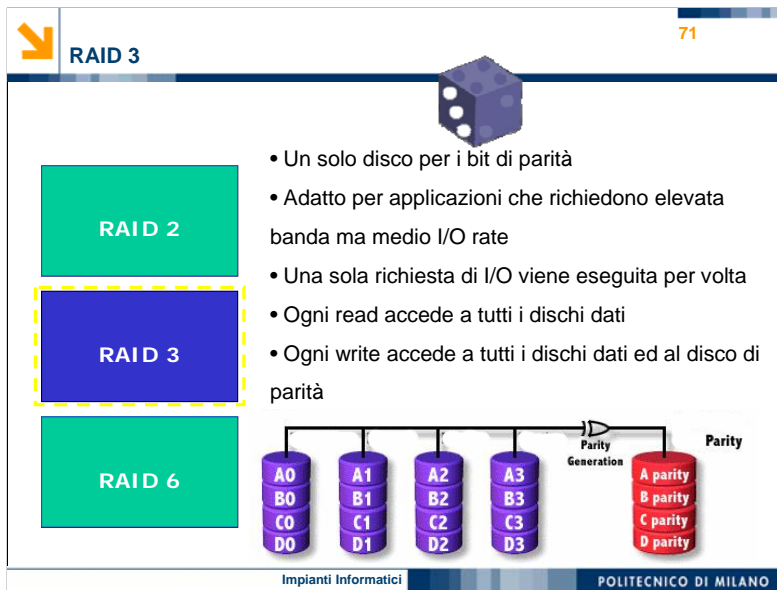


1. Il numero dei dischi ridondati è pari al
2. logaritmo in base 2 del numero di dischi contenenti i dati.
3. Questi dischi sono sincronizzati dal controllore, e data la sua complessità,
4. questo livello di RAID non è praticamente più in uso.



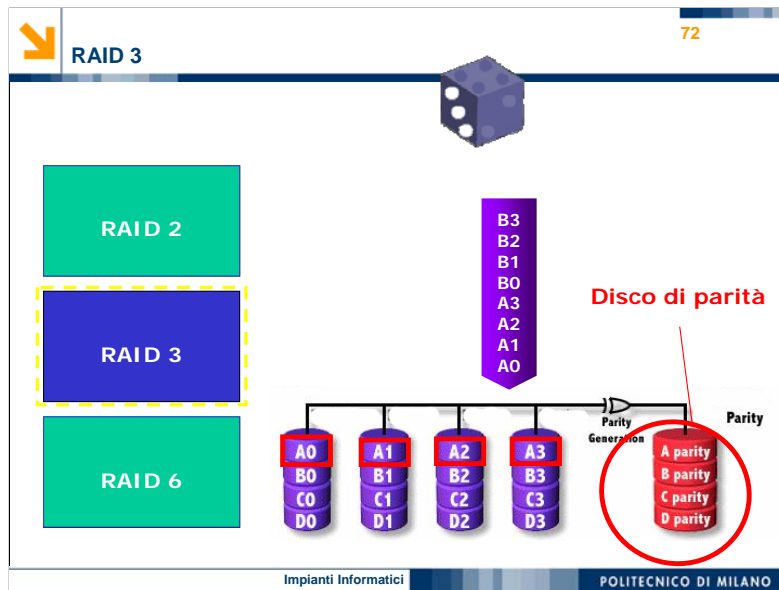
Il RAID di livello 3, implementa, al pari del livello 2,

1. Sia striping dei dati,
2. Che aggiunta di informazioni di ridondanza.
3. La principale differenza rispetto al livello 2 è che in questa implementazione i dati sono suddivisi a livello di byte
4. Analogamente al livello 2, l'affidabilità è buona,
5. Conseguita, anziché con i codici di hamming, mediante l'uso di informazioni di parità,.
6. Che contribuiscono, rispetto al livello 2, ad un maggiore sfruttamento della capienza fisica -.->
7. e della velocità del sistema,
8. In particolare per operazioni di I/O pesanti.




Nel RAID 3 le informazioni di ridondanza sono localizzate


1. in un singolo disco di parità, col rischio che diventi il bottleneck del sistema
Uno degli effetti collaterali del RAID-3 è che non può eseguire richieste multiple simultaneamente.
2. Questo perché ogni singolo blocco di dati è distribuito tra tutti i dischi del RAID, dato che lo striping è eseguito a livello di byte; così ogni operazione di I/O richiede di usare tutti i dischi.

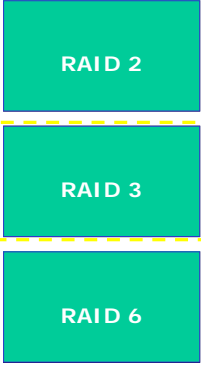


Nel RAID 3 le informazioni di ridondanza sono localizzate

1. in un singolo disco di parità, col rischio che diventi il bottleneck del sistema
Uno degli effetti collaterali del RAID-3 è che non può eseguire richieste multiple simultaneamente.
2. Questo perché ogni singolo blocco di dati è distribuito tra tutti i dischi del RAID, dato che lo striping è eseguito a livello di byte; così ogni operazione di I/O richiede di usare tutti i dischi.

 **RAID 6** 73





Striping

- Dati divisi a livello di *blocco*

Ridondanza

- Informazioni di parità
- Distribuita su tutti i dischi

Doppia ridondanza

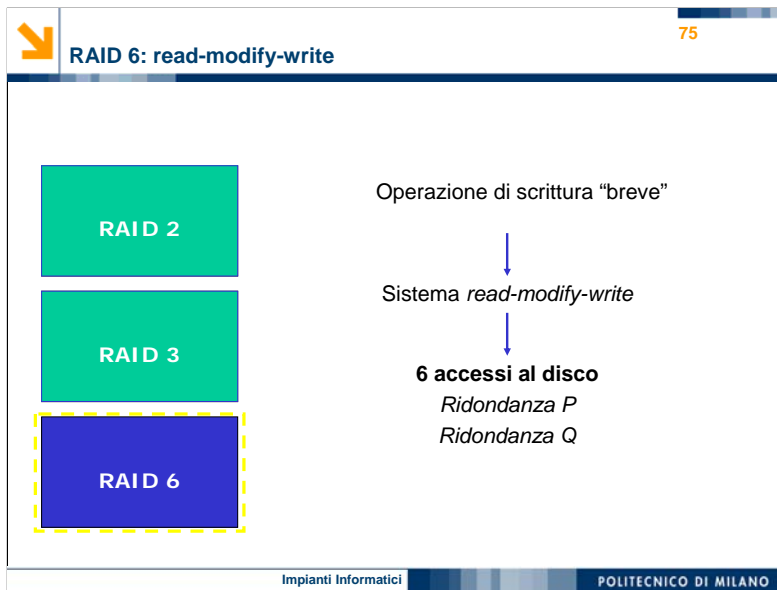
- Due parità indipendenti
- Alta affidabilità

Impianti Informatici POLITECNICO DI MILANO

Il RAID 6 è un'evoluzione del RAID 5.

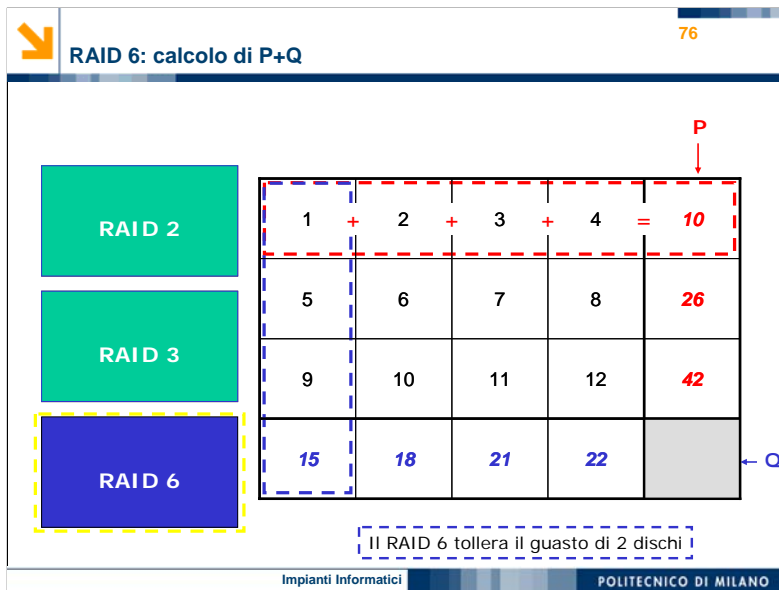
1. Analogamente implementa sia striping dei dati
2. A livello di blocco,
3. Sia l'uso di informazioni di parità per la ridondanza,
4. Anche in questa implementazione distribuite equamente tra tutti i dischi.
5. La differenza fondamentale è nell'uso di una doppia ridondanza
6. Mantenendo due stripe di parità indipendenti tra loro
7. E aumentando in tal modo l'affidabilità

1. $p+q$, in quanto viene mantenuta sia la ridondanza p
2. che la ridondanza q ,
3. Entrambe ripartite tra tutti i dischi dell'array



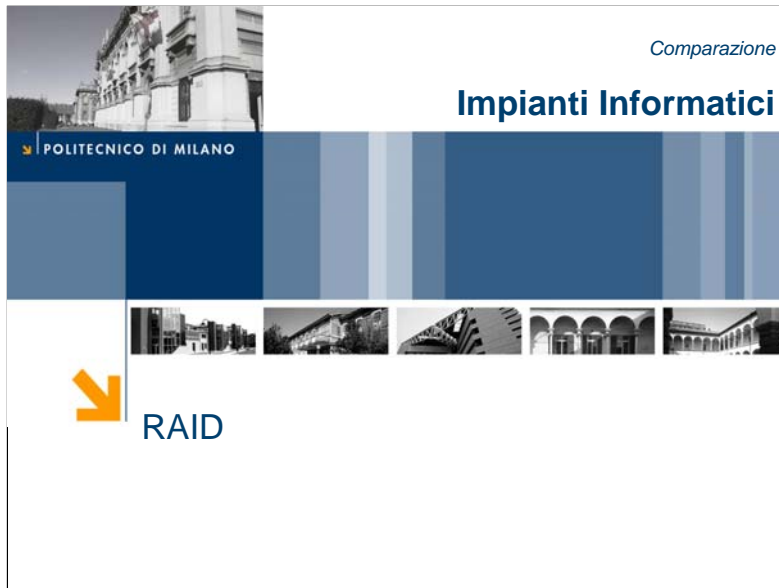
Lo schema p+q effettua

1. le operazioni di scrittura breve
2. utilizzando lo stesso procedimento **read-modify-write** dei livelli precedenti
Invece di quattro accessi al disco, come ad esempio nel RAID 5,
3. per ogni richiesta di scrittura si richiedono **sei** accessi al disco, a causa della necessità di leggere e aggiornare
4. sia la ridondanza "P", sia la ridondanza "Q".



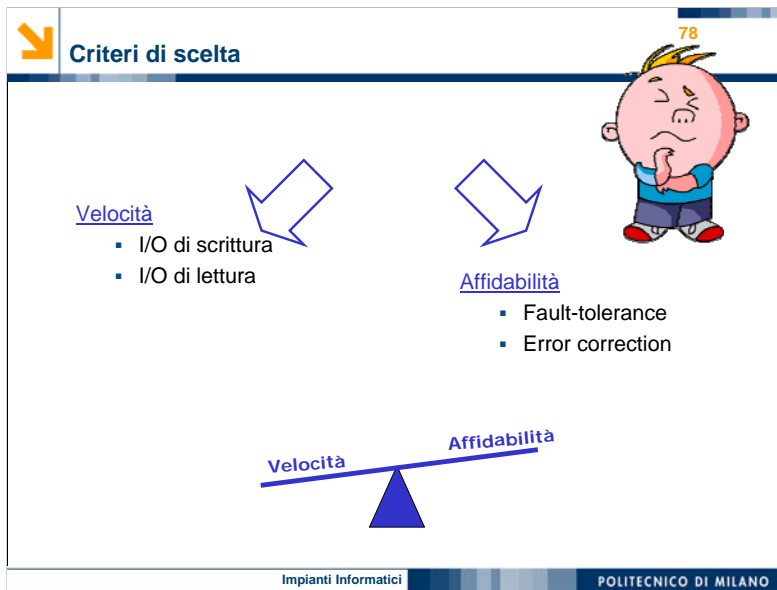
La tabella mostra con un esempio l'uso di una Doppia ridondanza,

1. qui calcolata come la somma dei vari blocchi
2. facenti parte le rispettive unità di stripe,
3. Una orizzontale, e
4. Una verticale. In questo modo
5. il RAID 6 è tollerante al guasto fino a due qualsiasi dischi dell'array



Hot Word:

- Mirroring (slide 9)
- RAID 0
- RAID 1
- RAID 0+1
- RAID 1+0



Vediamo ora alcuni criteri per scegliere l'implementazione di un sistema RAID piuttosto che un altro,

1. In particolare focalizzandosi sui parametri relativi alla velocità dell'array di dischi,
2. Tanto in scrittura
3. quanto in lettura
4. E all'affidabilità del sistema.
5. La scelta del livello di RAID sarà un compromesso tra questi due elementi



Velocità

- I/O di scrittura
- I/O di lettura
- tempi di recovery

Parallelismo

Affidabilità

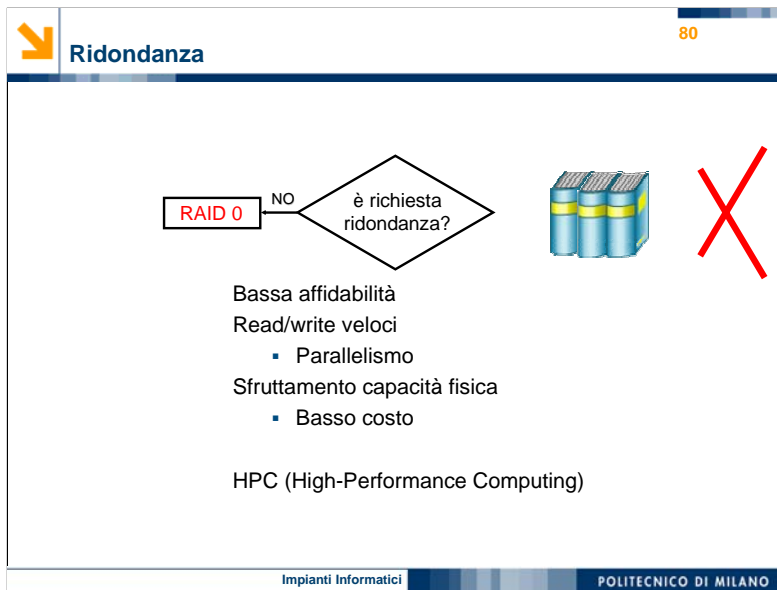
- Fault-tolerance
- correzione errori

Ridondanza

Duplicazione

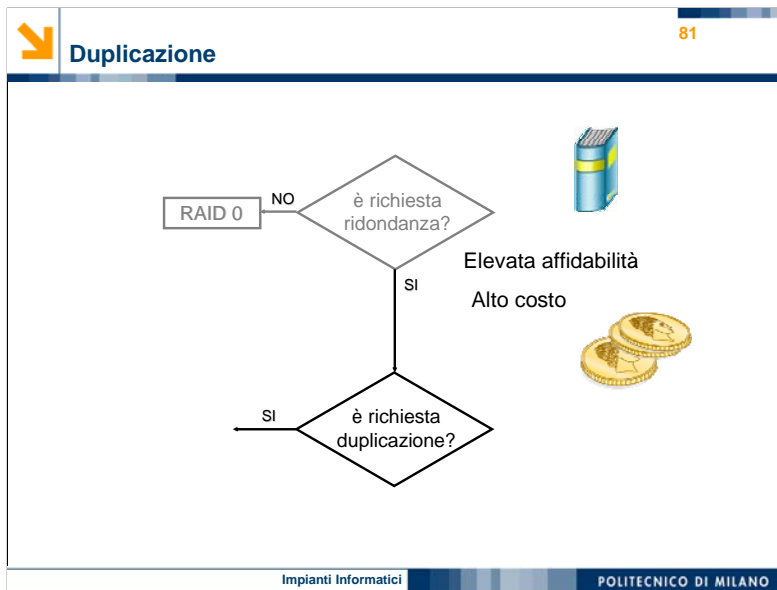
Costi

- sfruttamento della capacità fisica
- tipi di soluzione
- caratteristiche controller



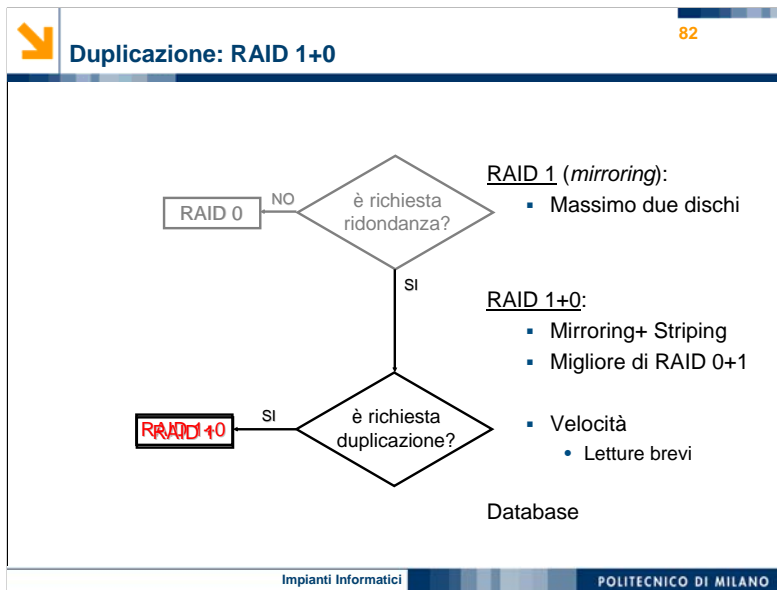
Un primo parametro di analisi riguarda il requisito di

1. avere informazioni ridondanti oppure no.
2. Se non è necessaria un'elevata affidabilità del sistema, e l'esigenza è in termini di
3. alta velocità del sistema,
4. con accesso parallelo alle risorse,
5. sfruttamento massimo della capienza
6. e implementazione a basso costo,
7. La scelta può ricadere sul più RAID di livello 0, che esegue il puro striping dei dati.
8. Il dominio in cui è indicato è, ad esempio, quello dell'high performance computing, in cui velocità e capacità sono più importanti dell'affidabilità



Se uno dei requisiti del sistema

1. è l'affidabilità
2. La scelta deve cadere in una soluzione che preveda la ridondanza dei dati.
3. Se l'affidabilità deve essere particolarmente elevata e
4. Non si hanno particolari requisiti economici,
5. Il mirroring rappresenta un giusto compromesso



1. L'uso del semplice RAID 1 è spesso limitativo in termini di capacità massima,
2. Dato che può implementare al massimo due dischi.
Si preferisce allora la sua combinazione con il RAID 0, in particolare la scelta
3. RAID 1+0 risulta generalmente migliore, sia dal punto di vista delle prestazioni, che della tolleranza ai guasti,
4. Rispetto al RAID 0+1.
5. Altro aspetto positivo della duplicazione è la velocità dell'array,
6. soprattutto in lettura e per brevi operazioni.
7. Un suo possibile dominio applicativo è quello dei database, dove è richiesto un alto transaction rate

➔
RAID 1+0
83

Affidabilità

- Fault-tolerant con 1 disco rotto
- Tollerante anche a rotture di più dischi, purché di differenti mirror

Prestazioni

- Meno efficiente di RAID 1 e RAID 0+1
- *Mirroring + Striping*

RAID 1+0

1
2
3
4

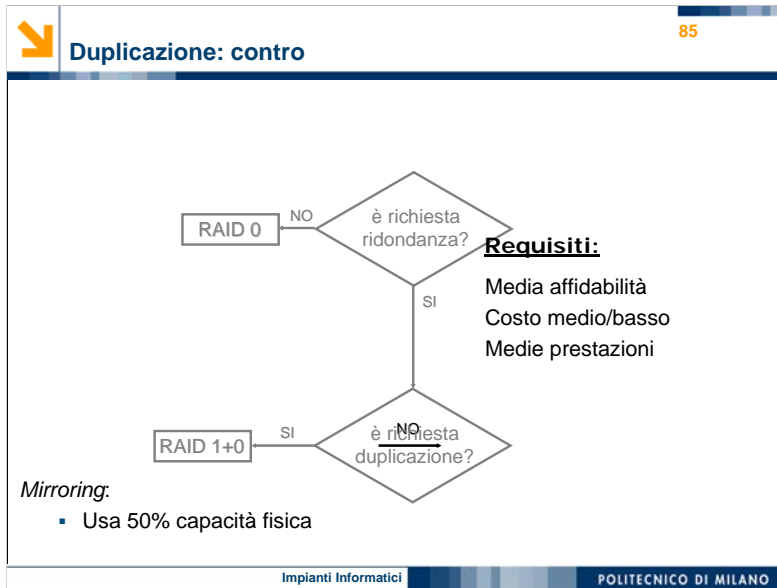
Impianti Informatici
POLITECNICO DI MILANO

Come già visto,

1. Il RAID 1+0 è tollerante alla rottura
2. di un disco, ma
3. Può sopportare anche la rottura di ulteriori supporti, purché non facciano parte di un mirror già coinvolto.
4. Dal punto di vista prestazionale,
5. Il raid 1+0, paragonato con il RAID 1 e con il RAID 0+1, sfrutta in misura minore l'accesso parallelo ai dati,
6. Dando priorità al mirroring rispetto allo striping dei dati.

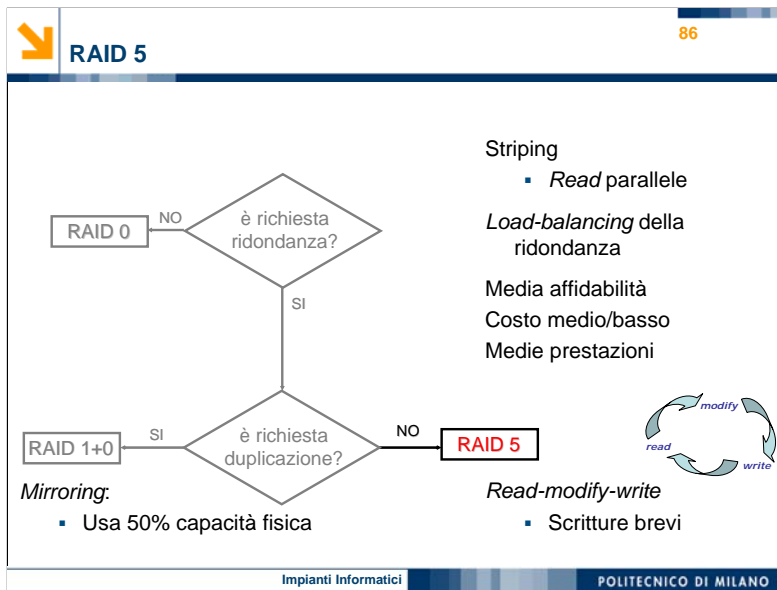


- La disposizione dei blocchi è identica se non per i dischi che sono in un diverso ordine
- Alcuni controller 0+1 combinano in un'unica operazione striping e mirroring
- **0+1**
 - non tollera due guasti simultanei (eccetto nel caso in cui interessino la stessa stripe)
 - nel caso di guasto a un singolo disco, qualunque guasto ad altra stripe è un *single point of failure*
 - il ripristino del disco richiede la partecipazione di tutti i dischi dell'array
- **1+0**
 - un disco per ogni gruppo RAID 1 può guastarsi ma se non riparato, l'altro disco è *single point of failure* dell'intero array



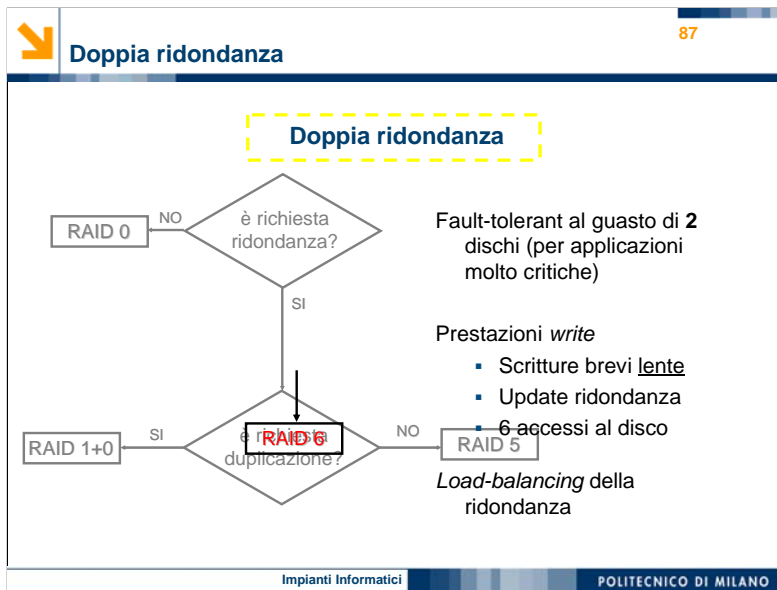
Non sempre

1. il mirroring rappresenta la migliore soluzione,
2. soprattutto a causa del suo alto costo di implementazione dovuto all'utilizzo di solo il 50% dello storage fisico
3. Altri livelli di RAID garantiscono un sufficiente compromesso
4. Tra affidabilità del sistema e
5. Costo di implementazione,
6. A volte con una leggera penalità nelle prestazioni



Uno tra i livelli più diffusi di RAID

1. è il 5, che implementa a livello di blocco
2. Lo striping dei dati, permettendo
3. la parallelizzazione degli accessi in lettura.
4. Anche le informazioni di ridondanza sono ripartite in modo bilanciato tra i dischi.
5. In scrittura il RAID 5 è gravato dal sistema read-modify-write,
6. Che penalizza le prestazioni delle write brevi



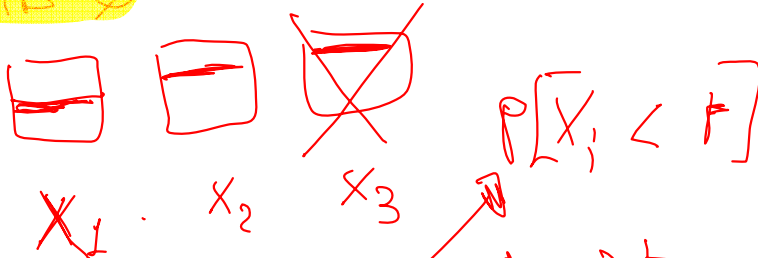
In sistemi e domini particolarmente critici potrebbe essere necessaria ancora una maggiore affidabilità dello storage,

1. Ad esempio con l'uso di una doppia ridondanza,
2. Che lo renda tollerante anche al guasto di due qualsiasi dischi dell'array.
3. La contropartita è un peggioramento delle prestazioni,
4. In particolare delle operazioni di scrittura leggere
5. Dato che l'aggiornamento della doppia ridondanza con il sistema read-modify-write
6. Richiede ben 6 accessi ai dischi.
7. Il RAID in questione è il 6, altrimenti detto P+Q.
8. Al pari del raid 5 le informazioni di parità sono equamente distribuite tra i dischi



$$- MTTF = MTDL$$

RAID 5



$$\rightarrow X_i \sim F(t) = 1 - e^{-\lambda t} \Rightarrow MTTF = \frac{1}{\lambda}$$

$$MTTF = MTDL \approx \min(X_i) = 1 - [1 - F(t)]^n \\ = 1 - e^{-\lambda n t} \Rightarrow MTTF = \frac{1}{\lambda n}$$



$$MTTF_n = \frac{MTTF_1}{n}$$

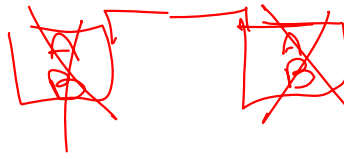
$$P[n \text{ dischi}] = n P[1 \text{ disco}]$$

$$1 - e^{-\lambda n t} \approx \lambda n t$$
$$= \frac{1}{MTTF_1} \cdot n \cdot t$$



$$1 - e^{-\lambda t} \Rightarrow \lambda = \frac{1}{MTTF} \quad | \approx \lambda t$$

90

RAID 1 $P[\text{loss of RAID 1 guasto}] =$

$$= P[1^o \text{ guasto}] \cdot P[2^o \text{ guasto} < \underline{MTTR}]$$

$$= \left(\frac{2 \cdot t}{MTTF_1} \right) \cdot \left(\frac{1}{MTTF_2} \cdot MTTR \right) \frac{MTTF_1^2}{2 \cdot MTTR}$$
$$= \frac{2}{MTTF_1^2} \cdot MTTR \cdot t \Rightarrow MTTR_{RAID 1} = \frac{MTTF_1^2}{2 \cdot MTTR}$$

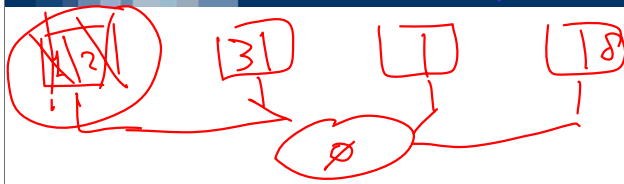
Impianti Informatici

POLITECNICO DI MILANO



RAID 1+0 (VEDI SLIDE 104)

91



$$P[1^{\circ} \text{ guasto}] \cong m \lambda t = \frac{m}{MTTF_1} \cdot t$$

$$P[2^{\circ} \text{ guasto nello stesso mirror} < MTR] =$$

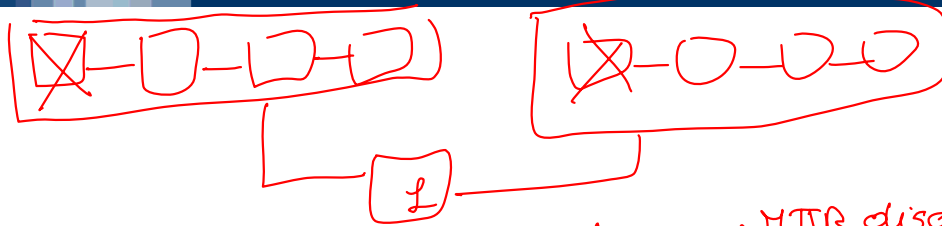
$$= \frac{1}{MTTF_2} \cdot MTR$$

$$\frac{m}{MTTF_1^2} \cdot MTR \cdot \cancel{P} = \cancel{P} \cdot MTR \cdot \underset{\substack{\text{RAD} \\ 1+0}}{1} = \frac{MTTF_2^2}{m \cdot MTR}$$



RAID $\phi+1$

92



$$P[1^{\circ} \text{ guasto}] \approx \frac{n}{MTTF_d} \cdot t$$

- MTTR disco
- RIAPISTINO RAID

$$P[2^{\circ} \text{ guasto nel mirror costante}] < \langle MTTR \rangle =$$

$$\frac{(n/2)}{MTTF_d} \cdot \text{MTTR} \Rightarrow MTTR_{RAID \phi+1} = \frac{2 \cdot MTTF_d^2}{n^2 \cdot MTTR}$$



$$MTTF_1 = 1000 \text{ gg} \approx 3 \text{ anni}$$

$$MTTR = 10 \text{ gg}$$

$$n = 8$$

$$MTDT_{1+\varnothing} = \frac{MTTF_1^2}{n \cdot MTTR} = 12500 \text{ gg} \approx 34 \text{ anni}$$

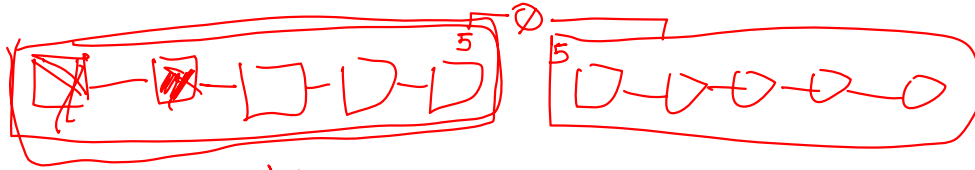
$$MTDT_{\varnothing+1} = \frac{MTTF_1^2 \cdot 2}{n^2 \cdot MTTR} = 3125 \approx 8 \text{ anni}$$



RAID 5

5+0

94



$M = \# \text{ dischi}$

$G = \# \text{ gruppi} \Rightarrow N = \# \text{ dischi} = \frac{M}{G}$
x gruppo

$\rightarrow P[\text{array } 5+0] \approx P[1 \text{ gruppo guasto}] \cdot G$

$P[1 \text{ gruppo guasto}] = P[1 \text{ disco in 1 gruppo guasto}] \cdot P[2^o \text{ guasto} < MTR \text{ in 1 gruppo}]$
 \downarrow
RAID 5



$$= \left(\frac{N}{M T F_1} \cdot t \right) \cdot \frac{(N-1)}{M T F_1} \cdot M T R = \frac{N(N-1)}{M T F_1^2} \cdot M T R \cdot t$$

$$r[\text{energy } 5+\phi] = G \cdot N(N-1) \cdot M T R \cdot t$$

$$M = G \cdot N \quad \Rightarrow \quad \frac{M \cdot (N-1)}{M T F_1^2} \cdot M T R \cdot t$$

$$\Rightarrow M T D \int_{5+\phi} = \frac{M T F_1^2}{M \cdot (N-1) \cdot M T R} = 100\%$$



$$\left. \begin{aligned} MTF_1 &= 1000 \\ MTR &= 10 \\ p &= \cancel{10} 8 \\ G &= 2 \\ N &= \cancel{8} 4 \end{aligned} \right\}$$

$$\begin{aligned} MTDL &= \frac{MTF_1^2}{M \cdot (N-1) \cdot MTR} \\ &= \frac{1000^2}{\cancel{10} \cdot \cancel{8} \cdot 3 \cdot 10} = \cancel{2500} \\ &= 4166.67 \end{aligned}$$



$G = ?$ (2)

$M = 50/100$
 $N = 5$

97

□ □ □ □ □ □ □ □ □ □

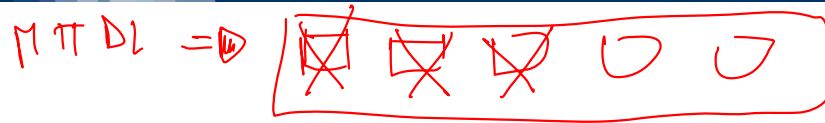
$$\begin{array}{l} M = 10 \\ G = 1 \\ N = 10 \end{array} \left\{ \begin{array}{l} \text{RAID 5} \\ \text{MTTDL} = \frac{\text{MTTF}^2}{10 \cdot 9 \cdot \text{MTTR}} \end{array} \right.$$

$$\begin{array}{l} M = 10 \\ G = 2 \\ N = 5 \end{array} \left\{ \begin{array}{l} \text{RAID 5+2} \\ \text{MTTDL} = \frac{\text{MTTF}^2}{10 \cdot 4 \cdot \text{MTTR}} \end{array} \right.$$



RAID 6 (P+Q)

98



$P[\text{RAID 6 guasto}] =$

$$P[1^{\circ} \text{ guasto}] \cdot P[2^{\circ} < \text{MTTR}] \cdot P[3^{\circ} < \text{MTTR}] =$$

$$= \frac{N}{\text{MTTF}_1} \cdot t \cdot \frac{(N-1)}{\text{MTTF}_1} \cdot \text{MTTR} \cdot \frac{(N-2)}{\text{MTTF}_1} \cdot \text{MTTR}$$

$$= \frac{N(N-1)(N-2)}{\text{MTTF}_1^3} \cdot \text{MTTR}^2 \cdot t$$



99

G gruppi
 M dischi

$$P[\text{RAID } G+1] =$$

$$= \frac{(G \cdot N) (N-1) (N-2)}{M \text{TF}_1^3} \cdot M \text{TR}^2 \cdot t$$

$$M \text{TDL}_{\text{RAID } G+1} = \frac{M \text{TF}_1^3}{M \cdot (N-1) (N-2) \cdot M \text{TR}^2}$$

$$M \text{TF}_1 = 1000 \text{ s}$$

$$M = 10 \quad N = 5$$

$$\Rightarrow M \text{TDL}_{G+1} = 83000 = 228 \text{ anni}$$



$$\bar{T}_{ACCESSO} = ?$$

100

	L	t [ns]	miss RATE	P	TT
Reg	1	1	10^{-1}	1	1
L1	2	1	$5 \cdot 10^{-2}$	10^{-1}	10^{-1}
L2	3	8	$4 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	$40 \cdot 10^{-3}$
RAM	4	100	$2 \cdot 10^{-2}$	$20 \cdot 10^{-5}$	$2000 \cdot 10^{-5}$
Local	5	10^7	$1 \cdot 10^{-2}$	$40 \cdot 10^{-7}$	40
NET	6	$5 \cdot 10^7$	$1 \cdot 10^{-2}$	$40 \cdot 10^{-9}$	200 2
Remote	7	$4 \cdot 10^8$	0	$40 \cdot 10^{-11}$	$160 \cdot 10^{-3}$
					ΣTT

Vecchie SIMM: 60/70ns
SDRAM/DDR: <10ns

10 ms
(3/4 ms)

Impianti Informatici

POLITECNICO DI MILANO



Caratteristiche indicative di dischi (2004)

101

Characteristics	Seagate ST373453	Seagate ST3200822	Seagate ST94811A
Disk diameter (inches)	2.50	3.50	3.50
Formatted data capacity (GB)	73.4	200	40.0
Cylinders	31310		
Sectors per drive	143,374,744	390,721,968 (LBA mode)	78,140,160 (LBA mode)
Number of disk surfaces (heads)	8	4	2
Rotation speed (RPM)	15,000	7200	5400
Internal disk cache size (MB)	8	8	8
External interface, bandwidth (MB/sec)	Ultra320 SCSI, 320	Serial ATA, 150	Ultra ATA, 100
Sustained transfer rate (MB/sec)	57-86	32-58	34
Minimum seek (read/write) (ms)	0.2/0.4	1.0/1.2	1.5/2.0
Average seek (read/write) (ms)	3.6/4.0	8.5/9.5	12.0/14.0
Mean time to failure (MTTF) hours	1,200,000@25 °C	600,000@25 °C	330,000@25 °C
Warranty (years)	5	3	-
Nonrecoverable read error per bit read	< 1 per 10 ¹⁵	< 1 per 10 ¹⁴	< 1 per 10 ¹⁴
Price in 2004 (\$/GB)	\$5	\$0.5	\$2.5



Esempio di calcolo del tempo **medio** di accesso al dato

	<i>layer</i>	<i>tempo t</i>	<i>miss rate m</i>	<i>prob. p</i>	<i>p x t</i>
1	Reg	1,00E+00	1,00E-01	1,00E+00	1,00E+00
2	L1	1,00E+00	5,00E-02	1,00E-01	1,00E-01
3	L2	8,00E+00	2,00E-02	5,00E-03	4,00E-02
4	Main mem.	1,00E+02	1,00E-01	1,00E-04	1,00E-02
5	Local disk	1,00E+07	2,00E-02	1,00E-05	1,00E+02
6	Net server	5,00E+07	2,00E-02	2,00E-07	1,00E+01
7	Remote server	4,00E+08	0,00E+00	4,00E-09	1,60E+00
tot					112,75

$p(i) = p(i-1) \times m(i-1)$ *probabilità di accesso al livello (i)*
 $p(i) \times t(i)$ *tempo di accesso al livello (i)*
tempo totale = $\Sigma p(i) \times t(i)$



	<i>layer</i>	<i>tempo t</i>	<i>miss rate m</i>	<i>prob. p</i>	<i>p x t</i>
1	<i>Reg</i>	1,00E+00	1,00E-01	1,00E+00	1,00E+00
2	<i>L1</i>	1,00E+00	5,50E-02	1,00E-01	1,00E-01
3	<i>L2</i>	8,00E+00	2,20E-02	5,50E-03	4,40E-02
4	<i>Main mem.</i>	1,00E+02	1,10E-01	1,21E-04	1,21E-02
5	<i>Local disk</i>	1,00E+07	2,20E-02	1,33E-05	1,33E+02
6	<i>Net server</i>	5,00E+07	2,20E-02	2,93E-07	1,46E+01
7	<i>Remote server</i>	4,00E+08	0,00E+00	6,44E-09	2,58E+00
tot					151,47

Miss rate: +10%



RAID 1+0

$n = \# \text{ DISCHI}$

modo a) $P[\text{ARRAY GUASTO}] = P[1^{\circ} \text{ GUASTO}] \cdot P[2^{\circ} \text{ GUASTO nello stesso mirror} | 1^{\circ} \text{ GUASTO}]$

$$\approx \frac{n}{MTTF_1} \cdot t \cdot \frac{1}{MTTF_2} \cdot MTTR$$

$$= \frac{n \cdot MTTR}{MTTF_1^2} \cdot t \Rightarrow MTDL = \frac{MTTF_1^2}{n \cdot MTTR}$$

modo b) $P[\text{ARRAY GUASTO}] = P[1 \text{ mirror guasto}] \cdot \# \text{ MIRROR (come RAID 5+0)}$

$$= \frac{2 \cdot MTTR \cdot t}{MTTF_1^2} \cdot \left(\frac{n}{2}\right) = \frac{n \cdot MTTR}{MTTF_1^2} \cdot t \Rightarrow MTDL = \frac{MTTF_1^2}{n \cdot MTTR}$$