

Collezione di slides presentate al corso di informatica Industriale

A. Fantechi

1. Duplicazione Configurabile
 - Esempi
 - Valutazione disponibilità
2. Dischi RAID
3. Diversity

- *Duplicazione configurabile*

- Una macchina *master* e una *slave (primary/secondary)*
- La macchina slave subentra a quella master in caso di guasto del master (*failover*)
- La rilevazione dell'errore avviene mediante codifica dell'errore o time-out: il fallimento è un crash della macchina, sempre rilevabile (*assunzione sui guasti*)
- con *riserva fredda (cold spare)*: lo slave viene tenuto normalmente spento
 - Problema della copia dei dati sensibili
- con *riserva calda (hot spare)*: lo slave funziona in parallelo al master, ricevendo gli stessi input; i suoi output non vengono forniti all'esterno.
 - la riserva calda permette di non attendere il tempo di start-up della riserva, e mantiene sempre una copia aggiornata dei dati sensibili.
 - richiedendo che anche la riserva sia sempre in funzione, diminuisce l'affidabilità complessiva

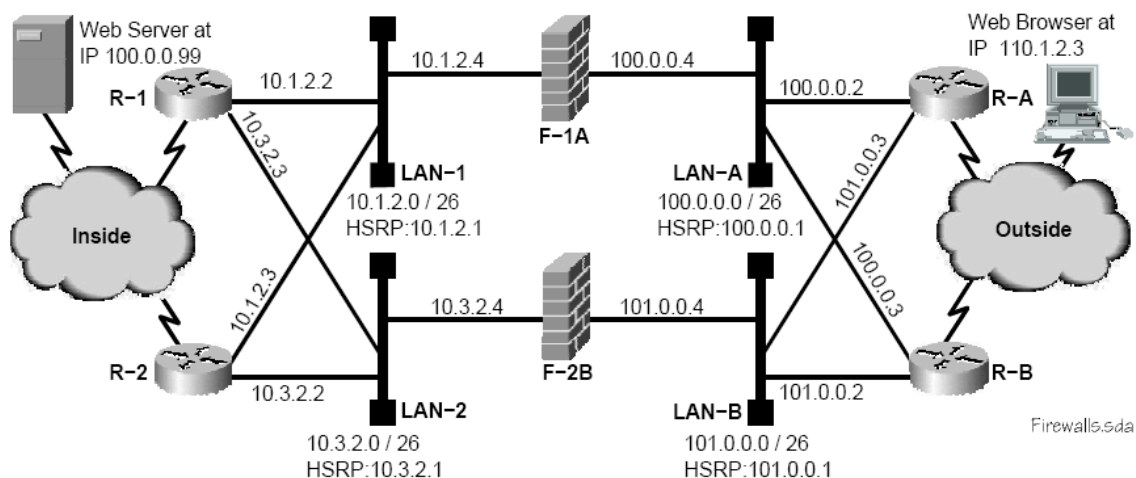
A configurable duplication solution from the open source world

- Build a redundant installation of OpenBSD firewalls against hardware failures. (OpenBSD is stable enough: no need to worry about software crashing, but concern about hard disks and other possibly failing hardware)
 - 1) Install a working firewall to be the "primary" firewall that operates continually under normal conditions.
 - 2) Install a "secondary" firewall with the same configuration, but have the interfaces set up with different addresses.
 - 3) Create a process on the "secondary" firewall to monitor the "primary" firewall.
 - 4) Install a software-controlled power switch on the primary firewall power line, and have it controlled by the secondary firewall.
- 5) Create a script on the secondary firewall to reassign the interfaces with the same addresses as the primary firewall, and to start the packet filter with the same rules as the primary firewall.

So, when the primary firewall fails, the secondary will kill the power to the secondary firewall, and reconfigure itself to run as the firewall for the network. Since the primary firewall is powered off, it can't interfere with the secondary firewall.

- → filtering function of firewall is assumed *stateless (static filtering)*: no need of recovering the state of the killed firewall

An industrial solution (Networking Unlimited)



Quattro routers e due firewall per coprire tutti i possibili guasti singoli
Riconfigurazione basata sull'aggiornamento automatico delle tabelle di routing a livello del protocollo IP.

Esempio di Valutazione disponibilità

- Consideriamo un sistema in cui un computer funzionante è affiancato da un altro identico che agisce come “riserva fredda”, subentrando nel funzionamento in caso di guasto del primo; l’interruzione del servizio dovuta allo start-up del computer freddo risulta essere di mezz’ora. (MTTR)
- Il tasso di fallimento di ognuno dei due computer è di $1 \cdot 10^{-5}$ ore (MTTF = circa 11 anni), e si suppone che il computer guasto venga rimpiazzato in tempo utile ad evitare un guasto doppio
- Come si valuta la disponibilità del sistema?

Esempio di Valutazione disponibilità

$$\begin{aligned} \text{Disp} &= \text{MTTF}/\text{MTBF} = \\ &= \text{MTTF}/(\text{MTTR}+\text{MTTF}) = \\ &= 100000/100000,5 = \\ &= 0,99999500002499\dots \end{aligned}$$

Non è “quasi uguale” a 1, ma l’indisponibilità è pari a 0,5 su 100000, cioè mezz’ora di mancato servizio ogni 11 anni (ovvio....), o circa 3 minuti di mancato servizio in un anno

→ (accettabile???) La disponibilità spesso si esprime in termini di “*nines*”,

cioè il numero di nove dietro lo zero

In questo caso si parla di una disponibilità di 5 *nines*

Guasti dei supporti di memorizzazione: Strutture RAID

- Uso di dischi in parallelo
 - Maggiori Prestazioni
 - Minore Affidabilità (l'uso di dischi in parallelo aumenta la probabilità di guasto)
- Affidabilità
 - Si migliora introducendo ridondanza
- Copiatura speculare (mirroring)

Dischi RAID

Redundant Arrays of Independent (Inexpensive) Disks

- molti dischi indipendenti visti come un unico grande disco logico ad elevate prestazioni
- i dati sono distribuiti (striped) su più dischi che sono acceduti in parallelo ottenendo:
 - **elevato transfer rate** per accessi a grandi quantità di dati (operazioni di I/O pesanti)
 - **elevato I/O rate** per accessi a piccole quantità di dati (operazioni di I/O leggere)
 - **load balancing** tra i vari dischi in modo automatico

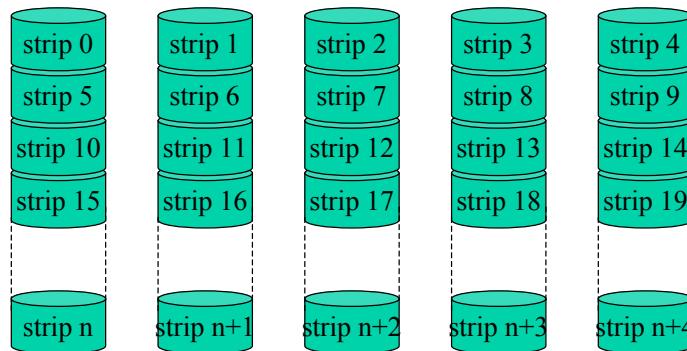
RAID: aumentata vulnerabilità

- array di 100 dischi: probabilità di guastarsi superiore di 100 volte quella di un disco solo
- se un disco ha un MTTF (Mean Time To Failure) di 200000 ore (~23 anni) un array di 100 dischi avrà un MTTF di 2000 ore (~ 3 mesi)
- **ridondanza** dei dati scritti: possibilità di correzione di eventuali errori/perdite di dati (disco guasto) con tecniche di codici a correzione di errore che utilizzano dati ridondanti scritti su dischi diversi da quelli sui quali vengono scritti i dati

data striping

- i dati sono distribuiti, in modo trasparente all'utente, su più dischi 'visti' come un unico disco veloce di grande capacità
- **striping**: suddivisione di dati che devono essere scritti sequenzialmente (un vettore, un file, ...) in segmenti (unità di stripe) che vengono scritti su più dischi fisici con un algoritmo round robin
- unità di stripe: quantità di dati che vengono scritti su un solo disco (~2÷128KB)
- **stripe width**: numero di dischi usati dall'algoritmo di striping (numero di dischi nell'array)

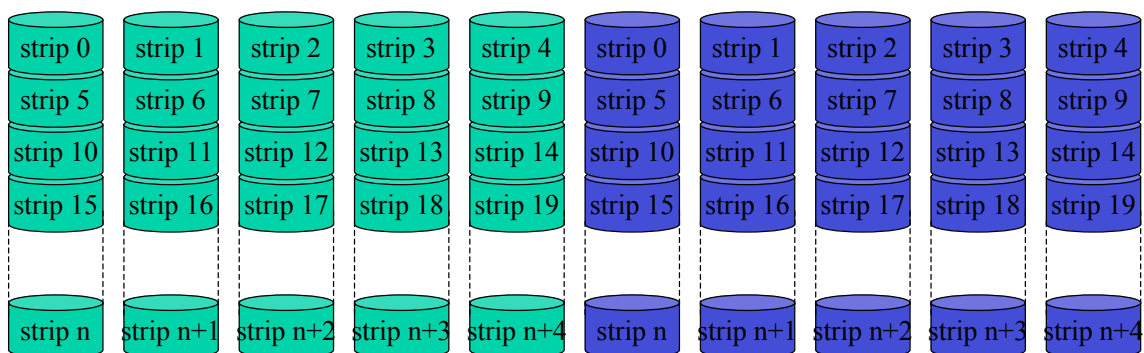
RAID-0



Le strip vengono scritte in parallelo:
maggiore velocità di accesso

Non c'è ridondanza

RAID - 1



copia primaria

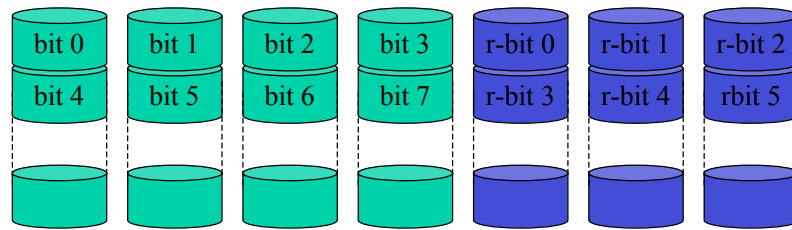
copia secondaria

Ridondanza (duplicazione) - mirroring

A seguito di un guasto, si può ripristinare
(una volta riparato/sostituito) il contenuto del disco guasto,
prelevandolo dalla copia

(si utilizza solo il 50% della capacità fisica)

RAID-2



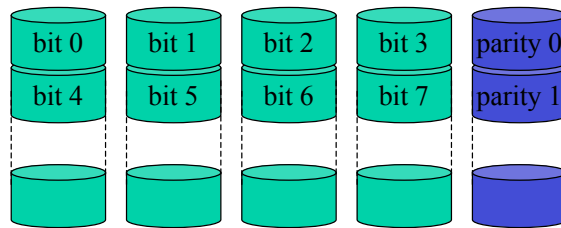
codifica dei dati usando un codice Hamming per ogni strip di dati. Questo codice può rivelare e correggere errori e permette il recupero dei dati senza una totale duplicazione.

(Codice Hamming)

Dimensione di parola	Bit di controllo	Dimensione totale	Percentuale assorbimento
8	4	12	50%
16	5	21	31%
32	6	38	19%
64	7	71	11%
128	8	136	6%
256	9	265	4%
512	10	522	2%

Numero di bit di controllo necessari per un codice di correzione per un solo errore (SEC: Single Error Correction)

RAID-3



La stessa word/byte e` distribuita su diversi dischi.

Il bit di parità consente di recuperare il contenuto di un disco guasto, e quindi ripristinare l'array una volta riparato il guasto.

Recupero del contenuto tramite parità

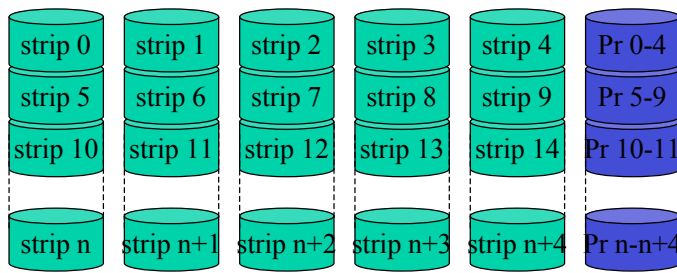
(somme modulo 2, cioè XOR)

- $\text{parity} = \sum_i \text{bit}_i$
- in caso di scrittura in cui il bit_i diventa bit'_i , basta eseguire la scrittura:
 $\text{parity} := \text{parity} + (\text{bit}_i - \text{bit}'_i)$, cioè:
 $\text{parity} := \text{parity} + \text{bit}_i + \text{bit}'_i$
- in caso di guasto al disco j , si recupera il valore del bit j -esimo eseguendo:

$$\text{bit}_j := \text{parity} - \sum_{i \neq j} \text{bit}_i, \text{ cioè:}$$

$$\text{bit}_j := \text{parity} + \sum_{i \neq j} \text{bit}_i$$

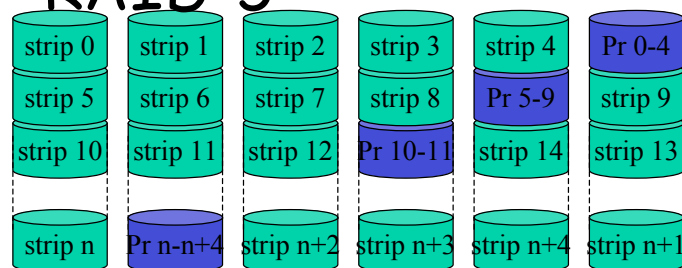
RAID-4



I dati sono distribuiti in blocchi, non in bit

Poiche` la parita` e` sullo stesso disco, tutte le transazioni richiedono un accesso ad esso, che quindi diventa un collo di bottiglia. La soluzione proposta dal RAID-5 consiste nel distribuire la parita` sui dischi

RAID-5



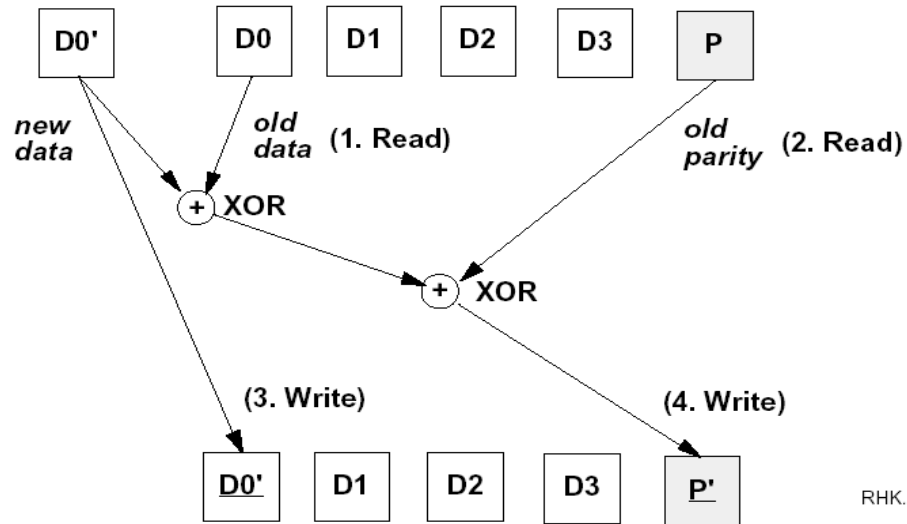
Soluzione più diffusa: maggiori vantaggi in termini di affidabilità e prestazioni, con minori costi per la bassa ridondanza

(Esiste anche un RAID-6 che e' un potenziamento del RAID-5: maggiore ridondanza per uso di codifica)

Problems of Disk Arrays: Small Writes

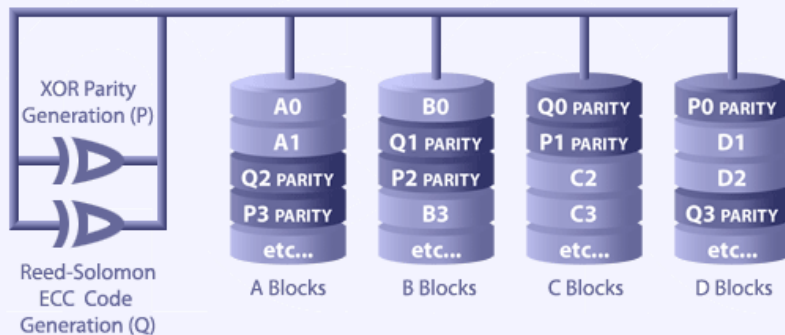
RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes



RHK.S96 15

RAID LEVEL 6 : Independent Data Disks with Two Independent Distributed Parity Schemes



Diversity

- **Luglio 1834, Dyonisius Lardner,**
“Babbage’s Calculating Engine”, Edinburgh review, n. CXX
“The most certain and effectual checks upon errors which arise in the process of computation, is to cause the same computations to be made by separate computers; and this check is rendered still more decisive if they make their computations by different methods.”
- **Dicembre 1837, Charles Babbage,**
“On the Mathematical Powers of the Calculating Engine” ,
unpublished manuscript, Museum of th eHistory of Science, Oxford
“When the formula to be computed is very complicated, it may be algebraically arranged for computation in tow or more totally distinct ways, and two or more set of cards may be made. If the same constants are now employed with each set, and if under these circumstances the results agree, we amy then be quite secure of the accuracy of them all.”

Diverse Programming

La **diversità** nella programmazione ha lo stesso scopo della ridondanza hardware: se si considerano gli errori software statisticamente distribuiti in un programma, l’uso di due (o più) versioni diverse di un programma permette di ovviare alla presenza di un errore.

Occorre evitare però che le due versioni diverse dello stesso programma possano presentare lo stesso errore.

Diversità di:

- Specifica
- Programmatore
- Algoritmo / Rappresentazione dei dati
- Linguaggio di programmazione
- Compilatore
- Sistema operativo
- Processore

Errori nella specifica

Errori di mancata comprensione delle specifiche

Errori nella soluzione scelta

Errori nel compilatore o nell’uso del linguaggio

Errori nel compilatore

Errori nel sistema operativo o nel suo uso

Errori del processore o nel suo uso

Richiedono ridondanza hardware

Use of Diverse Software

DEF-STAN 00-55

Software diversity techniques may be used for additional confidence in safety.

Where software diversity is used, the diverse components of the software will effectively be at a lower level of safety integrity compared to a non-diverse implementation and the rules of apportionment of integrity levels defined in Def Stan 00-56 shall apply.

It should be noted that the use of diverse software can introduce problems, for example in the areas of synchronization and comparison accuracies between diverse components operating in parallel, which may increase the safety risk.

Recovery Block

Se abbiamo due versioni diverse della stessa funzione:

```
int fvers1(int x,y);  
int fvers2(int x,y);
```

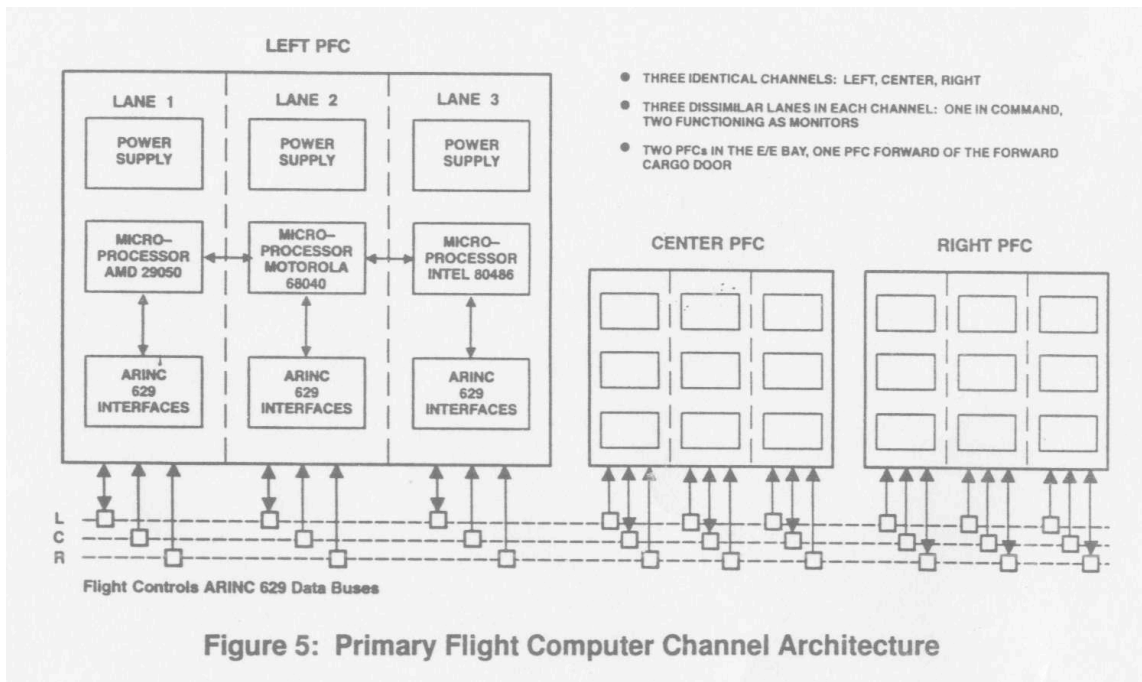
e un acceptance test (ad esempio test di ragionevolezza):

```
boolean acceptable(int x);
```

*Un **recovery block** si può realizzare come:*

```
int fsafe(int x,y);  
{int res;  
res = fvers1(x,y);  
if acceptable(res) return res;  
else {res = fvers2(x,y);  
      if acceptable(res) return res;  
      else failsafestate();  
}  
}
```

Boeing 777



Airbus A340

