Entrada / Salida
RAID

Agustín Fernández, Josep Llosa, Fermín Sánchez

Estructura de Computadors II
Departament d'Arquitectura de Computadors
Facultat d'Informàtica de Barcelona

Índice

Introducción

- La Industria ha adoptado un esquema estandarizado de múltiples discos en el diseño de bases de datos: RAID
  - RAID (Redundant Array of Inexpensive / Independent Disks)
- Objetivos
  - Aumentar el rendimiento (ancho de banda)
  - Aumentar la capacidad
  - Aumentar la fiabilidad (*reliability*) de los datos (tolerancia a fallos) en los sistemas de almacenamiento masivo (Discos Magnéticos)
- La idea detrás de RAID: Usar múltiples discos (más capacidad) que operen independientemente y en paralelo para incrementar el ancho de banda y mejorar la fiabilidad.
  - Ficheros distribuidos a través de múltiples discos. El ancho de banda aumenta con el número de discos
  - Se pueden añadir esquemas de redundancia y corrección de errores para mejorar la fiabilidad de los datos

http://www.storagereview.com/guide2000/ref/hdd/perf/raid/index.html

Introducción: Fiabilidad vs rendimiento

- Fiabilidad
  - La tecnología RAID protege los datos contra el fallo de una unidad de disco duro. Si se produce un fallo, RAID mantiene el servidor activo y en funcionamiento hasta que se sustituya la unidad defectuosa
  - Los sistemas RAID (excepto RAID 0) suponen la pérdida de parte de la capacidad de almacenamiento de los discos, para conseguir la redundancia o almacenar los datos de paridad
  - Los sistemas RAID profesionales deben incluir los elementos críticos por duplicado: fuentes de alimentación y ventiladores redundantes. De poco sirve disponer de un sistema tolerante al fallo de un disco si después falla por ejemplo una fuente de alimentación que provoca la caída del sistema
- Rendimiento
  - La tecnología RAID se utiliza también con mucha frecuencia para mejorar el rendimiento de servidores y estaciones de trabajo (se puede leer o escribir en varios discos simultáneamente)
- Estos dos objetivos, fiabilidad y mejora del rendimiento, no se excluyen entre sí
- RAID ofrece varias opciones, llamadas niveles RAID y numeradas desde 0 (RAID 0 – RAID 6). Cada opción proporciona un equilibrio distinto entre tolerancia a fallos, rendimiento y coste

Introducción: fiabilidad

- La fiabilidad se mide típicamente como tiempo medio hasta un fallo (MTTF: *Mean Time To Failure*)
  - Es el tiempo medio que transcurre hasta que un disco duro falla
  - Los discos actuales tienen MTTF de unas 50.000 horas (5,7 años de funcionamiento)
- ¿Qué ocurre en sistemas con múltiples discos?
- La seguridad de un conjunto (*array*) de discos está relacionada con el número de discos
  - fiabilidad de N discos = fiabilidad de 1 disco / N
  - MTTF de 1 disco = 50.000 horas = ≈5,7 años
  - MTTF de 8 discos en *array* = menos de 1 año
  - Servidores con más de 10 discos son usuales en el mercado
  - En algunas aplicaciones **no debe** haber problemas durante años
  - Añadir discos a un *array* para aumentar capacidad y ancho de banda reduce la fiabilidad
  - Se necesita redundancia en el *array*

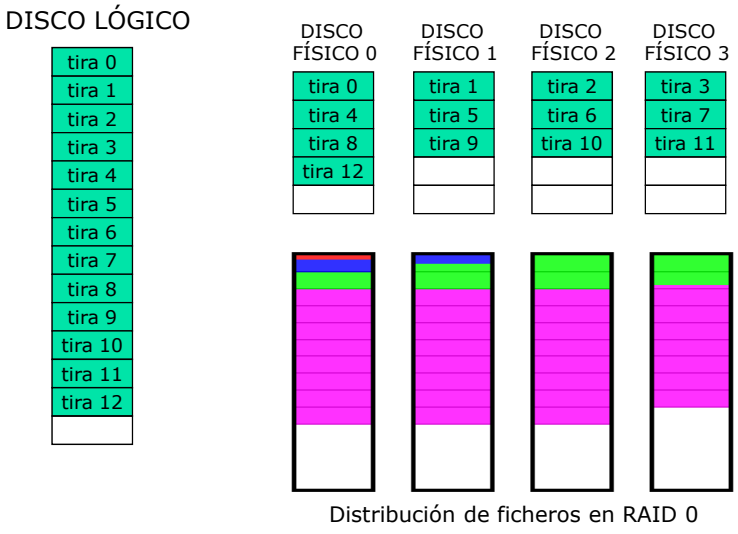
Introducción: conceptos básicos

- RAID es un conjunto de unidades físicas de disco vistas por el sistema operativo como una única unidad lógica.
- Los datos se distribuyen de forma entrelazada a través de las unidades físicas. Son posibles distintos niveles de entrelazado:
  - No entrelazado
  - Entrelazado a nivel de tira (*stripe*): cada fichero se divide en bloques llamados tiras que se distribuyen entre los discos. El tamaño típico de las tiras suele ser de 2 a 512 Kbytes
  - Entrelazado a nivel de byte
  - Entrelazado a nivel de bit
- La capacidad de los discos redundantes se usa para almacenar información que garantice la recuperación de los datos en caso de fallo del disco.
- Técnicas de redundancia de datos
  - No redundancia
  - *Mirroring*
  - Paridad
  - Códigos *hamming* horizontales
  - Códigos *Reed-Solomon*

RAID 0: Disk Striping

- Distribuye los datos con entrelazado a nivel de tira (*stripe*)
- También conocido como "separación o fraccionamiento/ *Striping*". Los datos se desglosan en pequeños segmentos y se distribuyen entre los discos del *array*
- Las unidades de disco conectadas en paralelo permiten una transferencia simultánea de datos a/de todos ellos, con lo que se obtiene una gran velocidad en las operaciones de lectura y escritura. Esto representa una gran ventaja en operaciones secuenciales con ficheros de gran tamaño
- Este nivel de RAID no ofrece tolerancia al fallo. Al no existir redundancia, RAID 0 no ofrece ninguna protección de los datos. El fallo de cualquier disco de la matriz tiene como resultado la pérdida de los datos y es necesario restaurarlos desde una copia de seguridad
- Este esquema es aconsejable en aplicaciones de tratamiento de imágenes, audio, video o CAD/CAM

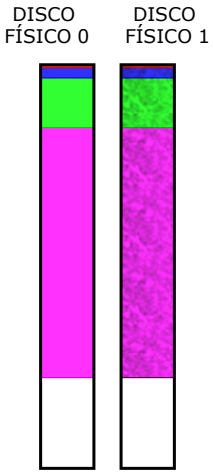
RAID 0: Disk Striping



Distribución de ficheros en RAID 0

RAID 1: Mirroring

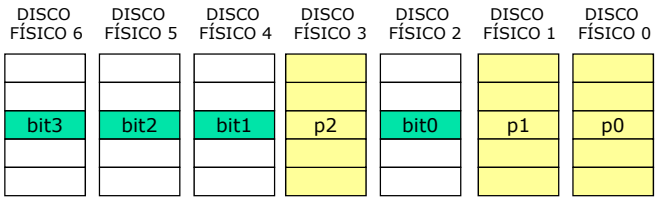
- *Mirror* = espejo
- Utiliza discos adicionales sobre los que se realiza una copia exacta de los datos
- Se duplican todos los datos. De esta manera se asegura la integridad de los datos y la tolerancia al fallo pues, en caso de avería, el controlador sigue trabajando con los discos no dañados sin detener el sistema
- RAID 1 ofrece una excelente disponibilidad de los datos mediante la redundancia total de los mismos.
- Los datos se pueden leer desde cualquiera de las copias
- Las escrituras son algo más lentas, se ha de escribir en las dos copias.
- RAID 1 es una alternativa costosa para los grandes sistemas, ya que duplica el coste de los discos



RAID 2: Redundancia a través del código *Hamming*

- Distribuye los datos con **entrelazado a nivel de bit**
- La operación E/S accede al mismo sector de todos los discos al mismo tiempo
- Adapta la técnica ECC (*Error Checking and Correction*), comúnmente usada para detectar y corregir errores en memorias de estado sólido
- El código ECC se intercala a través de varios discos a nivel de bit. El método empleado es el *Hamming*
- El código *Hamming* permite detectar y corregir 1 disco que falla o bien detectar que fallan 2 discos (pero no ambas cosas a la vez)
- Se usan códigos (14,4) o (39,7) con 14 y 39 discos respectivamente
- Prácticamente no se usa: con los discos actuales ya se sabe qué disco falla debido a que los discos incorporan la técnica ECC internamente a nivel de sector
- El resto de sistemas de corrección (RAID 3 a 6) se basan en tener información de qué disco/s falla/n

Ejemplo para un código (7,3)



Raid2: Códigos Hamming

- Ejemplo de código (7,3)

7	6	5	4	3	2	1	
b3	b2	b1	p2	b0	p1	p0	Palabra de 7 bits
b3	-	b1	-	b0	-	p0	p0 = b3 xor b1 xor b0
b3	b2	-	-	b0	p1	-	p1 = b3 xor b2 xor b0
b3	b2	b1	p2	-	-	-	p2 = b3 xor b2 xor b1

7	6	5	4	3	2	1
1	1	0	0	1	1	0

error en bit 5 !

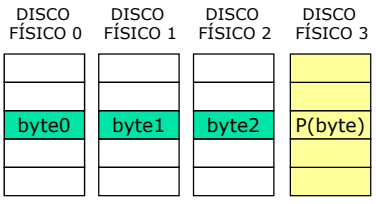
7	6	5	4	3	2	1
1	1	**1**	0	1	1	0

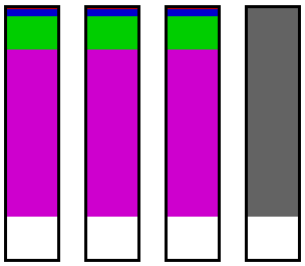
7	6	5	4	3	2	1	Test de paridad
1	1	**1**	0	1	1	0	
1	-	**1**	-	1	-	0	**T0 = p0** xor b3 xor b1 xor b0 = 1
1	1	-	-	1	1	-	**T1 = p1** xor b3 xor b2 xor b0 = 0
1	1	**1**	0	-	-	-	**T2 = p2** xor b3 xor b2 xor b1 = 1

  - El test de paridad da T2T1T0 = 101, indicando que el error está en el bit 5
  - Si da 000 es que no ha habido error
  - El sistema permite:
    - o bien corregir un error
    - o bien detectar 2 errores simultáneos, pero en ese caso apunta a un bit incorrecto

RAID 3: Acceso síncrono con un disco dedicado a paridad

- Distribuye los datos con **entrelazado a nivel de byte**
- Dedica un único disco al almacenamiento de información de paridad
- La información de ECC del disco (*Error Checking and Correction*) se usa para detectar errores. La recuperación de datos se consigue calculando la OR exclusiva (XOR) de la información registrada en los otros discos
- La operación E/S accede al mismo sector de todos los discos al mismo tiempo
  Su rendimiento de transacción es pobre porque todos los discos del conjunto operan al unísono





Distribución de ficheros en RAID 3

RAID 3: Acceso síncrono con un disco dedicado a paridad

- ¿Cómo se recuperan los datos?
  - Se usa la información ECC de cada disco para saber si un disco ha fallado
- Si tenemos que:

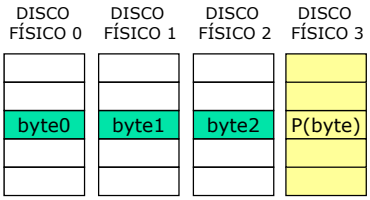
  P = X2 xor X1 xor X0

  Y por ejemplo se ha perdido el valor X1 (el disco 1 ha fallado).

  Se cumple que:

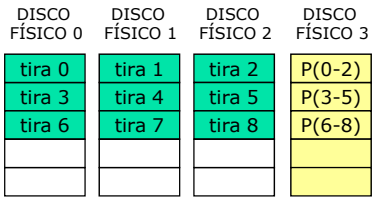
  X1 = P xor X2 xor X0

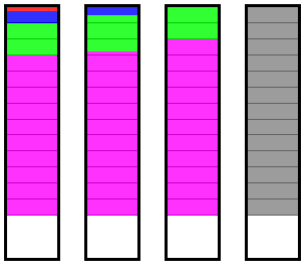
  - Esta operación se realiza bit a bit con los bytes de datos más el de paridad



RAID 4: Acceso Independiente con un disco dedicado a paridad

- Es similar a RAID 3 pero con **entrelazado a nivel de tira**
- En RAID 4 **se puede acceder a los discos de forma individual**.
- Basa su tolerancia al fallo en la utilización de un disco dedicado a guardar la información de paridad calculada a partir de los datos guardados en los otros discos. El disco de paridad es un cuello de botella del sistema
- En caso de avería de cualquiera de las unidades de disco, la información se puede reconstruir en tiempo real mediante la realización de una operación lógica XOR como en RAID 3, pero a nivel de tira



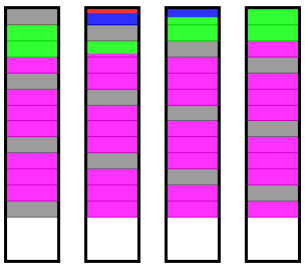


Distribución de ficheros en RAID 4

RAID 5: Acceso independiente con paridad distribuida

- Organizado de forma similar al RAID 4. La diferencia es que los bloques de paridad están distribuidos entre todos los discos
- Este esquema evita el cuello de botella que hay en el disco de paridad de RAID 4. Para ello, el RAID 5 no asigna un disco específico a esta misión, sino un bloque alternativo de cada disco. Al distribuir la función de comprobación entre todos los discos se disminuye el cuello de botella, y con una cantidad suficiente de discos puede llegar a eliminarse completamente, proporcionando una velocidad equivalente a un RAID 0

DISCO FÍSICO 0	DISCO FÍSICO 1	DISCO FÍSICO 2	DISCO FÍSICO 3
P(0-2)	tira 0	tira 1	tira 2
tira 3	P(3-5)	tira 4	tira 5
tira 6	tira 7	P(6-8)	tira 8



RAID 6: Acceso independiente con doble paridad

- Similar al RAID 5, pero incluye un segundo esquema de redundancia distribuido por los distintos discos y por tanto ofrece tolerancia extremadamente alta a los fallos y a las caídas de disco (dos niveles de redundancia)
- Hay pocos ejemplos comerciales en la actualidad, ya que su coste de implementación es mayor al de otros niveles RAID. Las controladoras que soportan esta doble paridad son más complejas y caras que las de otros niveles RAID.
- Las dos tiras redundantes son normalmente llamadas P y Q
  - P es la tira de paridad como en RAID 5
  - Q es el segundo nivel de redundancia basado en códigos Reed-Solomon
- Permite recuperar información aunque fallen hasta 2 discos
  - Para conocer mas detalles de cómo se calcula Q y cómo se corrigen 2 errores ver "The mathematics of RAID-6" http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf

DISCO FÍSICO 0	DISCO FÍSICO 1	DISCO FÍSICO 2	DISCO FÍSICO 3	DISCO FÍSICO 4	DISCO FÍSICO 5
tira 0	tira 1	tira 2	tira 3	P(0-3)	Q(0-3)
tira 4	tira 5	tira 6	P(4-7)	Q(4-7)	tira 7
tira 8	tira 9	P(8-11)	Q(8-11)	tira 10	tira 11

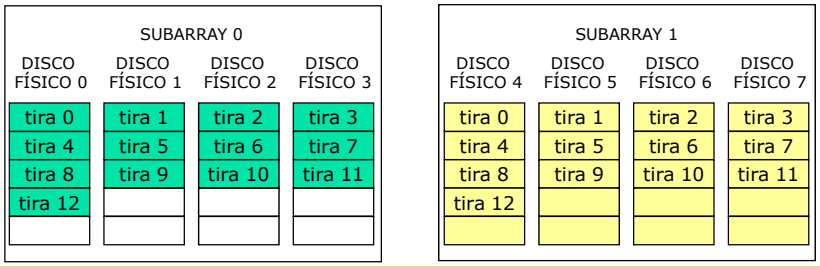
RAID: tabla resumen

Categoría	Nivel	Descripción	Grado de E/S solicitado (R/W)	Grado de transferencia de datos (R/W)	Aplicación típica
Acceso Independiente	0	Entrelazado de tira No redundante	Tiras largas: excelente	Pequeñas tiras: excelente	Aplicaciones que requieren altas prestaciones con datos no críticos
Estructura en espejo (*mirror*)	1	No entrelazado Duplicado	Bueno / regular	Regular / Regular	Controladores de sistemas, ficheros críticos
Acceso Paralelo	2	Entrelazado de bit Redundante con código *hamming*	Pobre	Excelente	
	3	Entrelazado de byte Redundancia con disco de paridad	Pobre	Excelente	Aplicaciones con mucha E/S, imágenes, CAD
Acceso Independiente	4	Entrelazado de tira Redundancia con disco de paridad	Excelente / regular	Excelente / pobre	
	5	Entrelazado de tira Paridad distribuida en bloques intercalados	Excelente / regular	Excelente / pobre	Grado de petición alto, lectura intensiva, consulta de datos
	6	Entrelazado de tira Paridad distribuida dual en bloques intercalados	Excelente / regular	Excelente / pobre	Aplicaciones que requieren alta disponibilidad

Niveles multi-RAID

- Los distintos niveles de RAID se pueden combinar entre ellos en pares para conseguir las ventajas de ambos niveles de RAID en un único sistema.
- Esquemas estandarizados multinivel
  - RAID 0+1 (RAID 01) y RAID 1+0 (RAID 10)
  - RAID 0+3 (RAID 03) y RAID 3+0 (RAID 30)
  - RAID 0+5 (RAID 05) y RAID 5+0 (RAID 50)
  - RAID 1+5 (RAID 15) y RAID 5+1 (RAID 51)
- El orden es importante: RAID X+Y ≠ RAID Y+X
  - RAID X+Y consiste en crear conjuntos de discos (subarrays) con RAID X y después tratar estos sub-arrays como discos individuales para crear un súper array con RAID Y
  - Ejemplo RAID 01

¿Cómo sería el resto de sistemas?. No todos son "razonables"



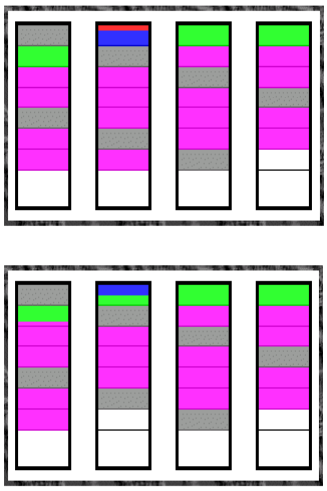
Niveles multi-RAID: RAID 0+3 y 3+0 (y 53)

- Is the most confusing naming of any of the RAID levels. In an ideal world, this level would be named RAID 0+3 (or 03) or RAID 3+0 (30). Instead, the number 53 is often used in place of 03 for unknown reasons, and worse, 53 is often actually implemented as 30, not 03.
- RAID 03 and 30 combine byte striping, parity and block striping to create large arrays that are conceptually difficult to understand.
- RAID 03 is formed by putting into a RAID 3 array a number of striped RAID 0 arrays.
- RAID 30 is more common and is formed by striping across a number of RAID 3 sub-arrays. The combination of parity, small-block striping and large-block striping makes analyzing the theoretical performance of this level difficult. In general, it provides performance better than RAID 3 due to the addition of RAID 0 striping, but closer to RAID 3 than RAID 0 in overall speed, especially on writes.
- RAID 30 provides better fault tolerance and rebuild performance than RAID 03, but both depend on the "width" of the RAID 3 dimension of the drive relative to the RAID 0 dimension: the more parity drives, the lower capacity and storage efficiency, but the greater the fault tolerance.

Niveles multi-RAID: RAID 0+5 y 5+0

- RAID 05 and 50 form large arrays by combining the block striping and parity of RAID 5 with the straight block striping of RAID 0.
- RAID 50 is a RAID 0 array striped across RAID 5 elements.
- RAID 05 is a RAID 5 array comprised of a number of striped RAID 0 arrays; it is less commonly seen than RAID 50.
- RAID 50 and 05 improve upon the performance of RAID 5 through the addition of RAID 0, particularly during writes. It also provides better fault tolerance than the single RAID level does, especially if configured as RAID 50.
- Most of the characteristics of RAID 05 and 50 are similar to those of RAID 03 and 30. RAID 50 and 05 tend to be preferable for transactional environments with smaller files than 03 and 30.

Niveles multi-RAID: Ejemplo de RAID 5+0



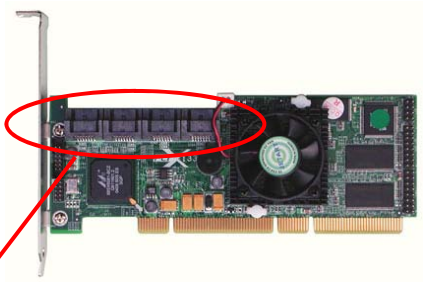
- Files of different sizes are distributed between the drives on an eight-disk RAID 5+0 array using a 16 KB stripe size.
- The red file is 4 kB in size; blue 20 kB; green 100 kB; and magenta 500 kB. Each vertical pixel represents 1 kB.
- Each of the large, patterned rectangles represents a four-drive RAID 5 array.
- Data are evenly striped between these two RAID 5 arrays using RAID 0.
- Then, within each RAID 5 array the data are stored using striping with parity. So, the first small file, and 12 kB of the second file, were sent to the top RAID 5 array; the remaining 8 kB of the second file and the first 8 kB of the 100 kB file went to the bottom RAID 5 array; then the next 16 kB of the 100 kB went to the top array, and so on.
- Within each RAID 5 array the data is striped and parity calculated just like a regular RAID 5 array; each array just does this with half the number of blocks it normally would.

Niveles multi-RAID: RAID 1+5 y 5+1

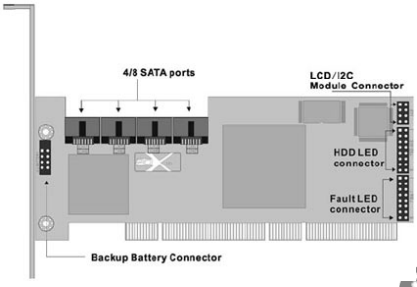
- Mirroring (or duplexing) combined with block striping with distributed parity.
- RAID 1+5 and 5+1 might be sarcastically called "the RAID levels for the truly paranoid". The only configurations that use both redundancy methods, mirroring and parity to maximize fault tolerance and availability, at the expense of just about everything else.
- A RAID 15 array is formed by creating a striped set with parity using multiple mirrored pairs as components; it is similar in concept to RAID 10 except that the striping is done with parity
- RAID 51 is created by mirroring entire RAID 5 arrays and is similar to RAID 01 except again that the sets are RAID 5 instead of RAID 0 and hence include parity protection.
- Performance for these arrays is good but not very high for the cost involved, nor relative to that of other multiple RAID levels.
- The fault tolerance of these RAID levels is truly amazing; an eight-drive RAID 15 array can tolerate the failure of any three drives simultaneously; an eight-drive RAID 51 array can also handle three and even as many as five, as long as at least one of the mirrored RAID 5 sets has no more than one failure! The price paid for this resiliency is complexity and cost of implementation, and very low storage efficiency.
- The RAID 1 component of this nested level may in fact use duplexing instead of mirroring to add even more fault tolerance.

Ejemplo comercial para PC

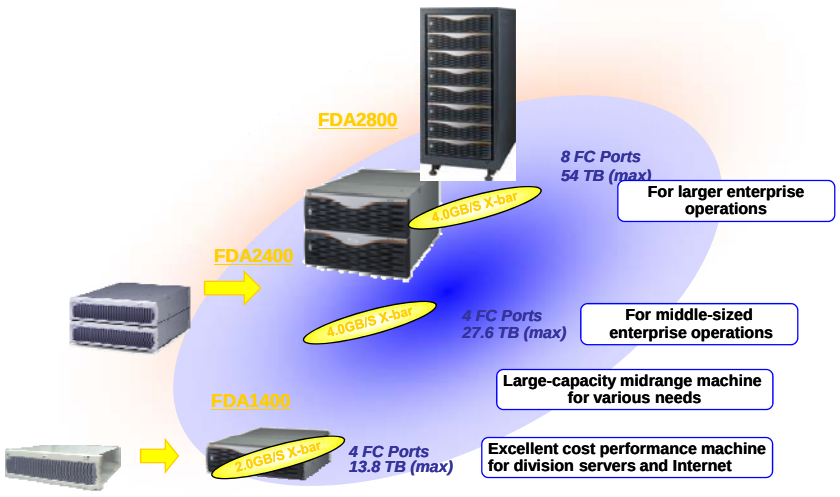
- Tarjeta controladora RAID
  - Conector bus PCI 64 bits
  - 8 ports SATA
  - Niveles RAID: 0, 1, 0+1, 3, 5, 6, JBOD
  - RAID Level Migration: downgrading only
  - Stripe Size Migration
  - Battery Backup Module optional



Ports SATA



Ejemplo comercial: FDA Product Range



Ejemplo comercial: NEC Storage S2800

Scalable, Best cost performance High-Mid range system

Specifications

- Capacity: **54 TB** (300GB HDD, RAID5 or RAID6), **240HDD**
- Server Interface: **8 Fibre Channel (2Gbps/1Gbps)**
- HDD Interface: **2Gbps Fibre Channel**
- Support HDD: **36GB (15krpm), 73GB (15k/10krpm), 147GB (15k/10krpm), 300GB (10krpm)**
- Support RAID: **RAID 1, 5, 6, 10, 50**
- Cache Capacity: **16 GB (8GB/Controller, Full Mirror)**

Availability

- Full Redundant
- RAID6, Hot Spare Disk
- Advanced error recovery technology “Phoenix”

Functions

- Storage Management: ***NEC StorageManager***
- Replication: ***DynamicDataReplication, RemoteDataReplication***
- Snapshot: ***DynamicSnapshot***
- Performance Monitoring: ***NEC Storage PerformanceMonitor***
- Access Control: ***AccessControl***
- CachePartitioning: ***CachePartitioning***