

Ejercicio II: Almacenamiento RAID

Nicolás A. Ortega Froysa

15 de octubre de 2021

1. Introducción: ¿Qué es RAID?

En la informática, aumentar las capacidades de una unidad suele ser más difícil que simplemente usar varias unidades. Por ejemplo, es más fácil que un ordenador use varios procesadores, que no construir un procesador más rápido. Esto es lo que lleva a que en el almacenamiento de información a largo plazo (i.e. en discos duros), para almacenar mucha información se suelen tener muchos discos. La dificultad con este modelo está en que se complica la organización de datos, ya que cada unidad se trata de forma separada. En esto, se han inventado varios sistemas lógicos de almacenamiento para gestionar el espacio en varios discos, como LVM. Pero la más conocida para almacenamiento masivo es **Redundant Array of Independent Disks**, o RAID.

El objetivo del almacenamiento con RAID es por seguridad, rendimiento, y capacidad de almacenamiento.

- **Seguridad:** se pueden hacer escrituras paralelas a dos discos de la misma información para poder tener copia de seguridad.
- **Rendimiento:** al distribuir información entre varios discos se puede paralelizar la transferencia de datos.
- **Capacidad de almacenamiento:** permite tratar a varios discos como si fueran un solo disco.

Aunque éstos son algunas de las ventajas de usar RAID, realmente depende del tipo de *nivel* que tenga.

2. Niveles RAID

Los sistemas configurados con RAID no funcionan todos igual, sino que se pueden configurar de forma distinta, usando distintos niveles, e incluso combinando éstos dependiendo de las necesidades. Dependiendo del nivel que sea, se explota una o más de las ventajas expuestas anteriormente.

2.1. RAID 0

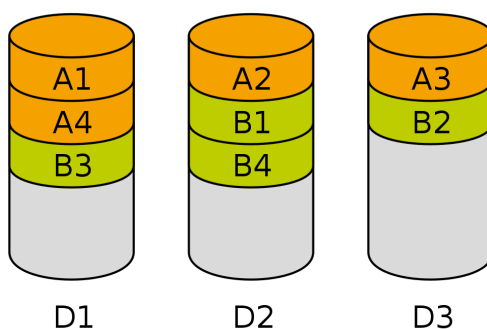


Figura 1: Striping con tres discos.

Para poder aprovechar tanto la mejora en rendimiento de tener varios discos, como su aumentada capacidad de almacenamiento, este nivel implementa una técnica que se denomina *striping*, tal que los segmentos consecutivos de un archivo se distribuyen entre los discos configurados (ver figura 1). Al usar varios discos, esto significa que la capacidad máxima de esta configuración RAID sería la suma del espacio de almacenamiento de todos sus discos. También mejora la transferencia de datos, al poder paralelizar su transferencia entre varios discos.

2.2. RAID 1

Para poder asegurarse de la seguridad de los datos guardados, este nivel automáticamente copia la información a los demás discos haciendo *mirroring*. Lo que hace es que mientras se escribe a un disco, se va escribiendo los mismos datos en el disco de copia. Esto no causa demasiado retraso, ya que las transmisiones son a dos discos distintos, lo cual se puede hacer en paralelo.

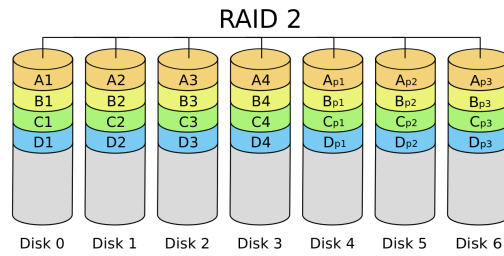


Figura 2: Diagram of RAID 2 setup.

2.3. RAID 2

Al igual que el RAID 0, hace *striping*, mas en vez de hacerlo a nivel de bloques, lo hace a nivel de bits. También se suma un método de corrección de errores con un *hamming code*. Mas debida a su complejidad comparada con un sistema de paridad, apenas se usa este nivel en la práctica. Esto hace que sea el único nivel original de RAID que no se usa actualmente. También es imposible de hacer varias peticiones a la vez por la distribución a nivel de bit en vez de ser por bloque, lo cual significa generalmente que para leer tan sólo un bloque ya debe de usar todos los discos.

2.4. RAID 3

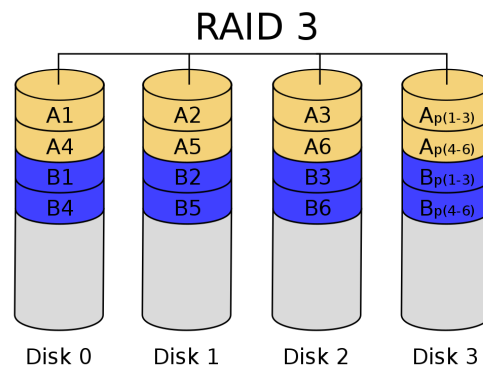


Figura 3: Diagram of RAID 3 setup.

Muy similar al nivel anterior, RAID 3 implemente *striping* a nivel de bytes, y un disco dedicado a guardar la paridad para la detección de errores. El hecho de que su *striping* sea a nivel de byte significa que, al igual que

en el nivel 2, no se pueden hacer varias peticiones a la vez. Esto significa que puede ser muy útil si se precisa una alta velocidad de transferencia de datos consecutivos de un mismo bloque. Mas este nivel ya casi no se usa en la práctica.

2.5. RAID 4

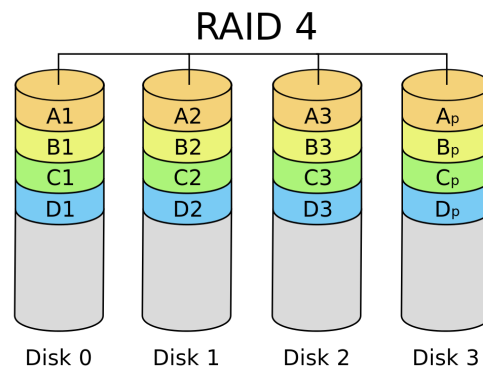


Figura 4: Diagram of RAID 4 setup.

Al igual que RAID 3, este nivel implementa *striping* y un disco dedicado a la paridad, pero a este nivel el *striping* sí que es a nivel de bloque, lo cual permite un acceso a datos más veloz, aunque la escritura pueda ser más lenta debido a la escritura al disco de paridad. Su ventaja es que puede ser fácilmente extendida siempre y cuando los nuevos discos estén completamente limpios (sólo bytes de ceros).

2.6. RAID 5

Funciona de manera igual que RAID 4, mas distribuye la información de paridad entre todos los discos, lo cual para funcionar tan sólo hace falta que funcionen todos los discos menos uno. Al faltar un solo disco, se pueden calcular los errores a partir de la información de paridad. Dicho lo cual, este nivel precisa de un mínimo de tres discos. Esto también mejora la velocidad de escritura respecto al RAID 4, ya que la información de paridad se encuentra también distribuida.

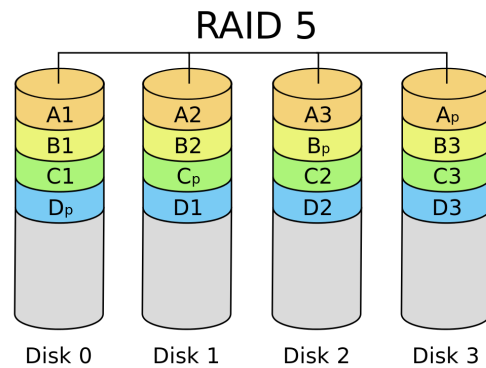


Figura 5: Diagram of RAID 5 setup.

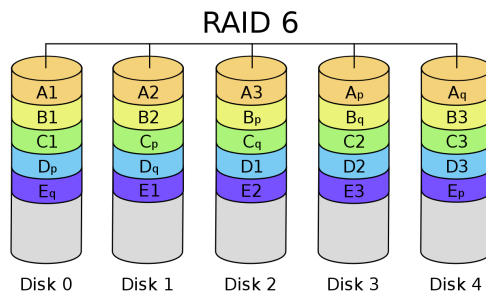


Figura 6: Diagram of RAID 6 setup.

2.7. RAID 6

Finalmente existe el nivel RAID 6. Se parece al RAID 5, en que también usa *striping* a nivel de bloques, y también usa paridad, pero en vez de ser sólo un bloque de paridad, son dos. Esto hace que haya más penalización al escribir debido a las paridades, pero aumenta la seguridad de los datos, ya que pueden fallar hasta dos discos.

3. RAID Híbrido (Nested)

Además de los niveles anteriores, también se pueden hacer combinaciones, que normalmente se denominan por *RAID X+Y*. Lo que significa esto es que a nivel más bajo está usando el nivel X, y a nivel más alto usa Y. De los más comunes, podríamos decir que existen:

- RAID 0+1 (01)

- RAID 0+3 (03)
- RAID 1+0 (10)
- RAID 5+0 (50)
- RAID 6+0 (60)
- RAID 10+0 (100)

4. Derechos de Autor

Copyright © 2021 Ortega Froysa, Nicolás <nicolas@ortegas.org>
Este documento está licenciado con la licencia Creative Commons Attribution No Derivatives 4.0 International.
Creado con L^AT_EX.