SISTEMAS DE FICHEROS

INTRODUCCIÓN

Parte del Sistema Operativo que administra la memoria de los dispositivos y unidades

Características:

- Ficheros identificados por un nombre
- Meta-información para cada fichero
    - Fecha de creación, permisos …
- Organización como una jerarquía tipo árbol

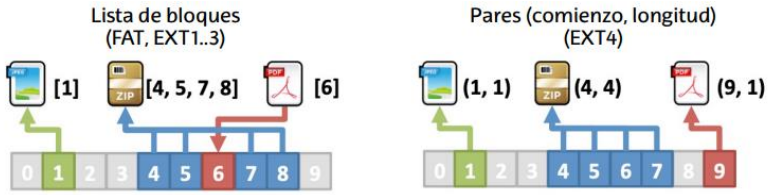
Implementación:

- Internamente los ficheros se almacenan en bloques secuenciales
- La jerarquía no se tiene en cuenta a nivel interno

Componentes de un fichero:

- Datos → 1+ bloques del disco con información binaria
- Meta-información → Nombre, tamaño, permisos, correspondencia con los bloques del disco

Cada fichero se guarda en al menos un bloque del disco:



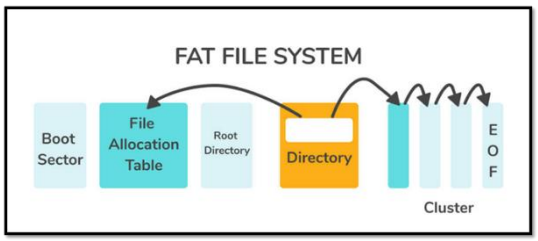
SISTEMA FAT

Creado en 1977 + utilizado por MS-DOS.

File Allocation Table (Tabla de reserva de ficheros) → Se basa en una lista enlazada que contiene la información de los bloques ocupados por cada archivo y los que están libres.

Características:

- Actualmente en desuso en sistemas de escritorio



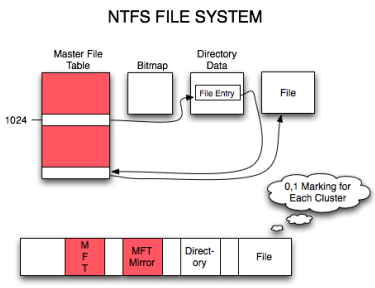
Variantes en uso:

- exFAT → Orientado a tarjetas flash (p.e. SDXC) y memorias USB
- FATX → Utilizado en las videoconsolas Xbox

SISTEMA NTFS

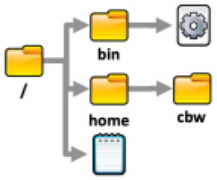
Creado en 1993 + utilizado en Microsoft Windows

New Technology File System → Introduce journaling: sistema de diario.



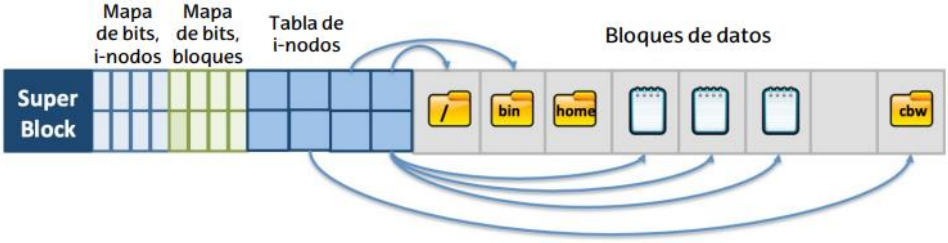
- Versión actual: v3.1 → No es tan compatible como FAT
- En proceso de ser sustituido por ReFS
    - Disponible en Windows Server desde 2012
    - Windows 11 puede leer discos ReFS (no crearlos)

SISTEMA EXT

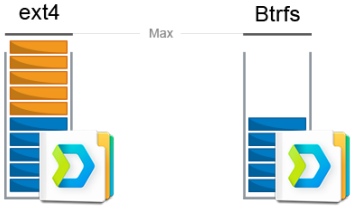


Creado en 1992, el primer sistema de ficheros para Linux → **Extended File System**

- Versión actual: Ext4 (en uso estable desde 2008)
- Compatibilidad hacia delante y hacia atrás → ext3 puede ser montado como ext4 sin cambios
- Organiza los datos en base a i-nodos



BtrFS



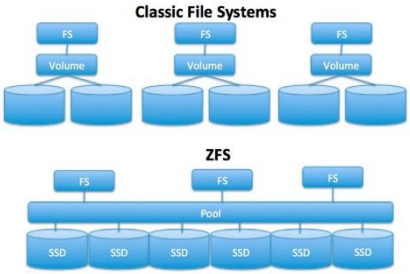
Creado en 2009, parte de Linux desde 2013.

Características

- Gestor de volúmenes integrado
- Capacidad de snapshots
- Recomendado para grandes tamaños de datos → El mismo BtrFS puede expandirse a varios discos

Posible sucesor de Ext4 → No parece que vaya a haber Ext5

ZFS

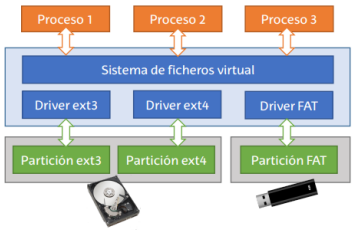


Creado en 2001 por Sun Microsystems, ahora Oracle

Sistema de ficheros y gestor de volúmenes

- Muy estable
- Licencia privativa → Polémica al respecto
- Variante libre: OpenZFS (2013)
    - Utilizado en entornos Linux

SISTEMA DE FICHEROS VIRTUAL



Interfaz de Linux:

- Expone una API POSIX a los procesos
- Envía las peticiones concretas al Driver que corresponda

Aunque el SSOO monte particiones de diferentes tipos, su uso es transparente a los procesos

ADMINISTRACIÓN

Respecto a los sistemas de ficheros, el sysadmin debe:

- Garantizar que los procesos de los usuarios pueden acceder a los sistemas de ficheros locales y remotos
- Supervisión y gestión de la capacidad de almacenamiento
- Gestionar copias de seguridad para evitar:
  - Corrupción de los datos
  - Errores hardware
  - Errores de usuario
- Garantizar la confidencialidad de los datos
- Conectar y configurar nuevos discos

Fichero del dispositivo

- Fichero del SSOO que posibilita que las aplicaciones accedan a un dispositivo (a través del kernel), p.e.:

cat /dev/dsp	Acceso a un DSP
cat /dev/input/mouse	Acceso al ratón

- Todos los dispositivos se encuentran en /dev:

Dispositivos SATA:	/sdXX	
Dispositivos RAID:	/mdX	
Especiales, p.e.	/null (nulo)	/urandom (números aleatorios)

Driver del dispositivo

- Rutinas del kernel que definen cómo se comunica el SSOO con el dispositivo: Interrupciones, DMA, …

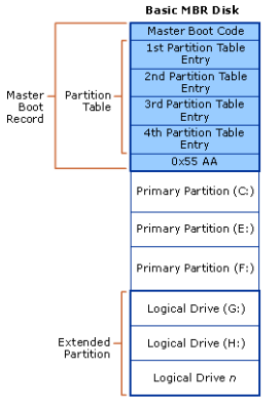
Partición

- Unidad de almacenamiento lógico que permite tratar un único dispositivo físico como varios
  - Un sistema de ficheros diferente en cada uno
- Utilidad:
  - Impide que ciertos directorios crezcan indefinidamente, p.e.:
    - /var/spool Para aplicaciones de colas (correo, impresión, …)
    - /tmp Para archivos temporales
  - Dividir el espacio para software y para archivos de usuarios
- En sistemas Unix/Linux:
  - Se encuentran en /dev, con un número adjunto al nombre del disco.

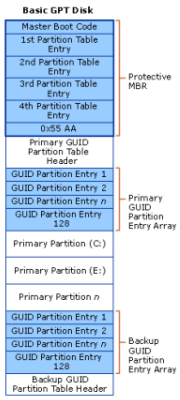
/dev/sdb	Disco
/dev/sdb1	Partición 1 del disco
/dev/sdb2	Partición 2 del disco

- Con Kernels recientes, el sistema crea un alias para cada partición:
  - Se puede usar cada vez que sea necesario
  - Evita tener que comprobar nombres después de cada reinicio
  - Listar UUID de cada partición: comando **blkid**

Tabla de particiones



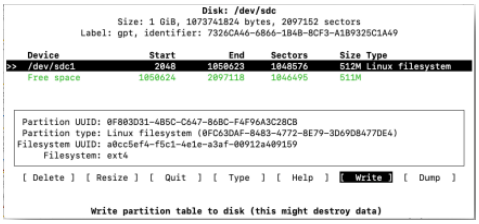
- Esquema de cómo se organizan las particiones del disco → Generalmente MBR o GPT
- **Master Boot Record (MBR)**
  - A veces mostrado como DOS (por MS-DOS) → 1983
  - Permite dividir 1 disco en 4 particiones primarias
  - Para crear más particiones:
    - Convertir 1 partición primaria en lógica
    - Crear particiones extendidas dentro la lógica
    - Límite: 23 particiones extendidas
  - Almacena la *meta información al comienzo* del disco
  - Límite: 2 TB por disco
- **GUID Partition Table (GPT)**
  - Permite realizar hasta 128 particiones en 1 disco (1990~2000)
  - Almacena la *meta-información distribuida* por el disco.
  - Límite: 9.7 zetabytes por disco
  - No es tan compatible como MBR
    - Para poder usar un disco GPT para arranque, el sistema debe ser BIOS UEFI.



TAREAS DE ADMINISTRACIÓN

Manipular la tabla de particiones:

- **Comando fdisk**
  - Sintaxis: fdisk <fichero-de-dispositivo> → P.e. fdisk /dev/sda
  - Algunas opciones:
    - p Ver tabla de particiones de disco
    - n Nueva partición
    - w Escribir nueva tabla de particiones
    - q Salir
- **Comando cfdisk**
  - Variante visual de fdisk
  - Más fácil de utilizar
  - No tiene todas las funciones de fdisk



Formatear una partición:

- Tras crear una partición, crear un sistema de ficheros en ella
  - Tipos de sistema soportados en /proc/filesystems
- **Comando mkfs**: crea un sistema de ficheros en una partición
  - Sintaxis: **mkfs.<tipo-de-sistema> <partición>** → P.e. mkfs.ext4 /dev/sda3
  - Forma alternativa: **mkfs [-V -t tipo-de-sistema] <partición>** → P.e. mkfs –t ext4 /dev/sda3
    - Se desaconseja el uso de esta forma

Montar una partición

- Habilitar el acceso al dispositivo desde el sistema de ficheros (usando el fichero de dispositivo)
- **Comando mount**
  - Sintaxis: mount <opciones> [fichero-disp] [punto-montaje]
  - Algunas opciones:
    - -t → Tipo de sistema
    - -r → Montar en sólo lectura
  - Ejemplo: mount -t ext3 /dev/sdc1 /home/unai/miDisco

Desmontar una partición

- **Comando umount**
- Sintaxis: umount [punto-montaje]
- Requiere que ningún proceso esté usando la partición
- Se puede usar el **comando lsof** para mostrar qué procesos la están usando

Montaje automático

- El fichero /etc/fstab define los dispositivos a montar automáticamente en el arranque del sistema
- Columnas:

‹file sys›	Dispositivo
‹mount point›	Punto de montaje (directorio)
‹type›	Tipo de partición: ext3, ext4, swap, ...
‹dump›	Frecuencia de backup, no se usa en la actualidad
‹pass›	Flag: ejecutar fsck al siguiente arranque

```
# /etc/fstab: static file system information.
#
#<file sys>  <mount point>   <type>      <options>          <dump>
<pass>
proc         /proc          proc        defaults           0       0
/dev/sda1    /              ext3        errors=remount-ro  0       1
/dev/sda5    none           swap        sw                 0       0
/dev/hdc     /media/cdrom0  udf,iso9660 user,noauto        0       0
/dev/fd0     /media/floppy0 auto        rw,user,noauto     0       0
```

Explorar las particiones del sistema

- **Comando lsblk**
  - Lista el hardware de almacenamiento y particiones
  - Parámetro “-e7” para ocultar particiones Snap en Ubuntu
- **Comando df**
  - Lista particiones y puntos de montajes
  - -h → mostrar tamaños en formato “humano”
  - -t → mostrar los tipos de sistemas de ficheros
- **Comando mount**
  - Muestra las particiones montadas
  - -l → listar
  - -t → indicar tipo de sistema, p.e. “-t ext4”

Comprobar el sistema de ficheros

- **Comando fsck**
  - Detección y corrección (no siempre) de problemas de corrupción en el sistema de ficheros
  - Compara la lista de bloques libres con las direcciones en los i-nodos
  - Verifica la lista de i-nodos libres con los i-nodos de los directorios
  - No es muy efectivo para detectar ficheros corruptos
- **Comando badblocks**
  - Detecta y excluye sectores inválidos del disco
  - Funciones SMART de un disco duro
    - Self Monitoring Analysis and Reporting Technology
    - Herramientas para acceder a la información de estado del disco
    - Software y funcionalidades dependen del fabricante

Redimensionar el sistema de ficheros

- **Comando resize2fs**
  - Versión del kernel >= 2.6
  - Espacio suficiente para poder redimensionar
  - Conveniente hacer una copia de seguridad de la tabla de particiones:
    - Utilizando dd: **dd if=/dev/sdc of=part.bkp count=1 bs=1.**
- **Comando parted**
  - Sintaxis: parted /dev/sdX
  - Permite copiar, mover, cambiar sistemas de ficheros

DISCOS EN GCP

Añadir un disco a una instancia

- Editar la configuración de la MV
- "Additional disks" → "Add new disk"
- Configurar el nuevo disco, entre otras:
  - Nombre y tamaño en GB
  - Origen: "blank disk" para un disco en blanco
  - Tipo: ver siguiente diapositiva
- Crear disco y guardar cambios en la instancia

Administrar discos

- Apartado "Discos" en la sección
- "Almacenamiento" de Compute Engine.

Tipos de disco

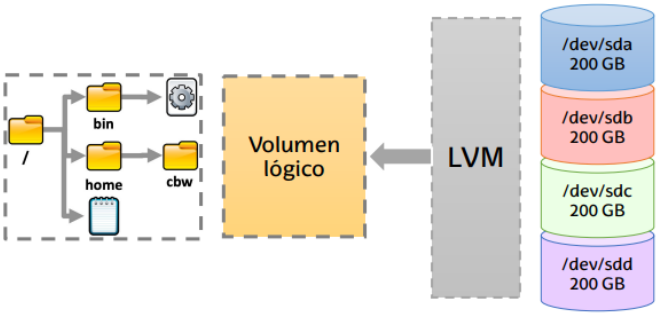
El coste mensual depende de la región

Tipo	Descripción	Coste ($/mes)
pd-standard	Discos duros tradicionales (los más lentos)	0.044 / GB
pd-balanced	Discos sólidos (SSD) configurados para ser competitivos en coste	0.110 / GB
pd-ssd	Discos sólidos (SSD)	0.187 / GB
pd-extreme	Discos sólidos (SSD) configurados para máximo rendimiento	0.137 / GB

LVM

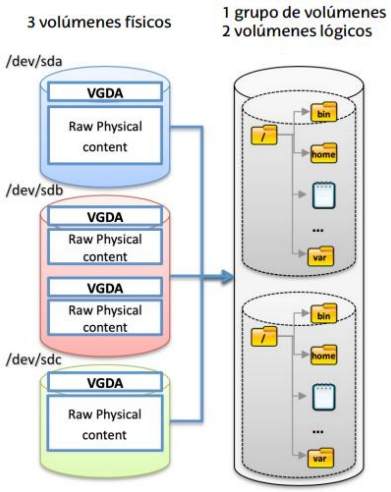
El sistema de ficheros ocupa 800 GB pero sólo tengo discos de 200 GB:

Logical Volume Manager (LVM) crea una capa de abstracción sobre el almacenamiento físico → Permite crear volúmenes lógicos que "escondan" el hardware real



Jerarquía

- Volúmenes físicos
  - Partición completa
  - Contiene el VGDA
  - Volume Group Descriptor Area
  - Contiene los datos físicos
- Grupos de volúmenes
  - Equivalente a "super-discos"
- Volúmenes lógicos
  - Equivalente a "super-particiones"
  - Albergan los sistemas de ficheros



Ventajas

- Gestión flexible del almacenamiento en disco
  - Elimina los límites del espacio físico
- Almacenamiento redimensionable
  - Los volúmenes se pueden agrandar/reducir de forma simple
  - Algunas operaciones no requieren desmontar el sistema de ficheros
- Traslado de datos en caliente
  - Los datos se pueden mover entre discos aunque estén en uso
  - Se puede reemplazar un disco sin interrumpir el servicio
- Captura de instantáneas
  - Simplifica las copias de seguridad

Administración

- **Comando pvcreate**
  - Crear un volumen físico
  - Sintaxis: pvcreate [partición]
  - Es necesario crear antes la partición (p.e. con cfdisk)
- **Comando vgcreate**
  - Crear un grupo de volúmenes con varios volúmenes físicos
  - Sintaxis: vgcreate [nombre-grupo] [vols-físicos]
  - Ejemplo: vgcreate grupovol /dev/sdb1 /dev/sdc1
- **Comando lvcreate**
  - Creación de un volumen lógico
  - Sintaxis: lvcreate [nombre-grupo] -l [tamaño] -n [nombre-volum-log]
  - Ejemplo: lvcreate grupovol -l 100%FREE -n miVolumen
- **Comando vgextend**
  - Añadir un nuevo volumen físico al grupo de volúmenes
- **Comando lvextend**
  - Extender un volumen lógico a un grupo de volúmenes mas grande
- **Comando resize2fs**
  - Redimensionar el sistema de ficheros
- **Comandos vgreduce (grupo de volúmenes) y lvreduce (volumen lógico)**
  - Para reducir el tamaño de los volúmenes
- Se mostrar el estado del volumen con **lvdisplay**

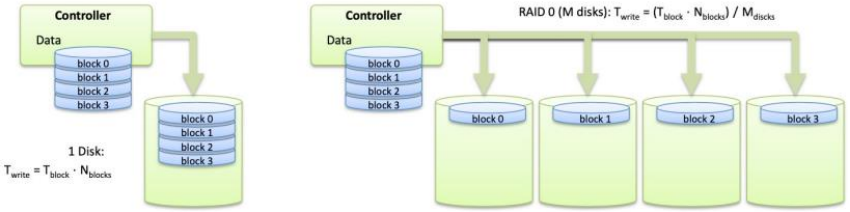
RAID

Redundant Array of Independent Disks

- Técnica de almacenamiento: los datos se distribuyen o replican entre varios discos
  - Transparente para el usuario y para el SO
- Diferentes opciones de configuración (niveles)
  - Según necesidades de fiabilidad, rendimiento y capacidad
- Se puede implementar a nivel HW o SW
  - Hardware: más eficiente pero más caro
    - Imagen: controladora PCI para RAID 0, 1, 5 y 10
  - Software: apropiado para RAID 0 y 1

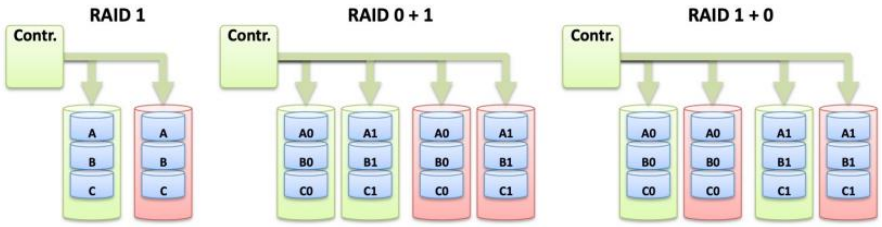
RAID 0: Striping (Volumen dividido)

- Los datos se dividen en segmentos y se distribuyen entre los discos
- Rendimiento: Bueno, acceso paralelo a los discos
    - Más discos → Más velocidad
- Fiabilidad: No hay tolerancia a fallos
- Capacidad: 100% de uso (0 redundancia)



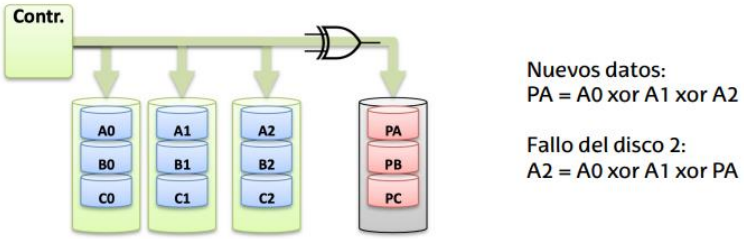
RAID 1: Espejo

- Utilizar un disco secundario para copiar todos los datos
- Rendimiento: Bajo, debido al exceso de escrituras
- Fiabilidad: Alta por la alta redundancia
- Capacidad: 50% de la disponible

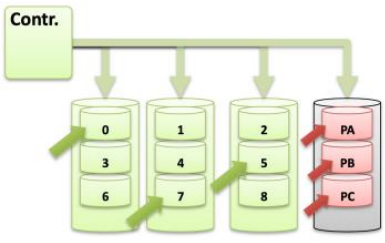


RAID 4: Striping + Paridad

- Un disco almacena información de paridad sobre el resto
- Rendimiento: Bueno en lectura, malo en escritura
- Fiabilidad: Tolerancia al fallo de 1 disco
- Capacidad: 1 disco dedicado exclusivamente a redundancia

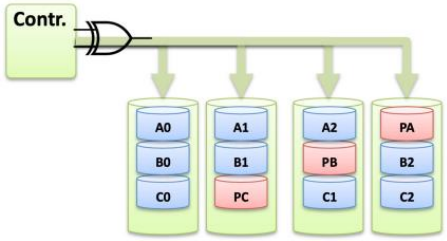


- El mayor problema en RAID 4 son las escrituras serializadas en el mismo disco
    - Ejemplo: actualizar las posiciones 0, 5 y 7
        1) Leer bloques 0, 5 y 7 y PA, PB y PC
        2) Calcular el nuevo valor de PA, PB y PC
        3) Escribir los nuevos bloques de datos
        4) Escribir los nuevos bloques de paridad
            a. Este último paso implica escrituras serializadas
            b. Bajo rendimiento



RAID 5: Striping + paridad distribuida

- La información de paridad se distribuye por todos los discos
- Rendimiento: Mejor que RAID 4, elimina la escritura serializada
- Fiabilidad: Tolerancia al fallo de 1 disco
- Capacidad: Se dedica a redundancia el equivalente a 1 disco



Otros niveles de RAID

- RAID 2, RAID 3
  - Paridad a nivel de bit (RAID2) o byte (RAID3), en lugar de bloque.
  - No es muy utilizado
- RAID 6: Striping + Doble paridad
  - RAID 4 pero usando el doble de espacio para paridad
  - Tolerante al fallo de 2 discos
- RAID anidados: jerarquías en árbol
  - P.e, RAID 0+1, RAID 1+ 0 (10), …

ADMINISTRACIÓN DE RAID

Comando mdadm:

- Creación de un dispositivo RAID
  - Para crear el dispositivo /dev/md0:
    - **mdadm --create /dev/md0 --verbose --level=0 --raid-devices=2 /dev/sdb /dev/sdc2**
  - Los discos tienen que haber sido previamente particionados (p.e. con cfdisk)
  - El proceso de creación se puede monitorizar:
    - cat /proc/mdstat
- Monitorizar el sistema RAID
  - **mdadm --monitor [opciones] /dev/md0**
- Eliminar (desactivar) RAID:
  - Parar el dispositivo: **mdadm --stop /dev/md0**
  - Limpiar información: **mdadm --zero-superblock /dev/sdX**
    - Limpia la información existente de un dispositivo RAID parado

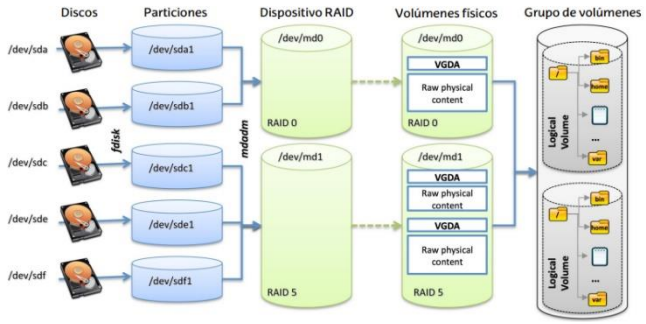
En caso de fallo de un disco (asumiendo un sistema RAID 5), el disco roto se puede recuperar automáticamente:

- Eliminar el disco roto del RAID:
  - **mdadm /dev/md0 –r /dev/sdc1**
- Reemplazar el disco físico por otro (debe ser idéntico)
- Crear particiones como en el original:
  - **fdisk /dev/sdc**
- Añadir al dispositivo RAID:
  - **mdadm /dev/md0 –a /dev/sdc1**
- Monitorizar el proceso de reconstrucción:
  - **cat /proc/mdstat**

- Se puede simular el fallo de un disco:
    - **Utilizar: mdadm/dev/md0 -f/dev/sdc1**
    - **Toda la información en: /var/log/syslog**

COMBINANDO RAID Y LVM

LVM se debe implementar sobre RAID



COPIAS DE SEGURIDAD

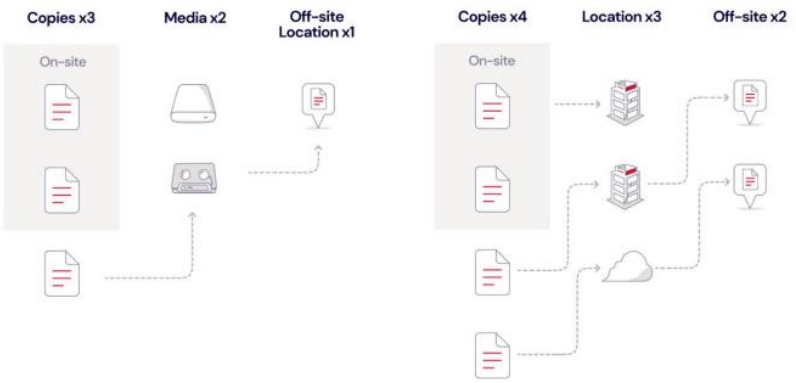
RAID + Journaling no es suficiente para tener una disponibilidad del 100%

- Tener copias de seguridad es esencial
    - Solucionar eventos inesperados, tanto HW como SW
    - Evita potenciales problemas de los usuarios
- Implica dedicar recursos exclusivos
    - Recursos físicos
        - Discos dedicados exclusivamente a copias, Servidores SAN, …
        - Cintas: LTO (LinearTape-Open), SAIT, AIT
    - Almacenamiento en la nube

La política debe ir acorde a los requisitos:

- Qué guardar
    - Datos de usuarios / aplicaciones / sistema
    - Partes críticas del sistema
- Cuándo hacer las copias
    - No recargar el sistema en momentos críticos
    - Dependerá del nivel de uso y la parte del sistema de ficheros
    - Automatizar las copias (usando p.e. cron)
- Dónde hacer las copias
    - Balance entre copias locales y en ubicaciones remotas

Estrategias:



Los discos pueden fallar/romperse/…

MFG	Model	Drive Size	Drive Count	Avg. Age (months)	Drive Days	Drive Failures	AFR
HGST	HMS5C4040ALE640	4TB	3,621	83.2	326,504	4	0.45%
HGST	HMS5C4040BLE640	4TB	11,934	80.1	1,083,231	22	0.74%
HGST	HUH728080ALE600	8TB	1,115	62	99,279	9	3.31%
HGST	HUH728080ALE604	8TB	90	71.1	8,094	3	13.53%
HGST	HUH721212ALE600	12TB	2,606	44.8	232,974	-	0.00%
HGST	HUH721212ALE604	12TB	13,203	27	1,181,748	42	1.30%
HGST	HUH721212ALN604	12TB	10,537	50.7	941,603	164	6.36%
Seagate	ST4000DM000	4TB	17,899	91.9	1,607,828	167	3.79%
Seagate	ST6000DX000	6TB	883	98.3	80,411	3	1.36%
Seagate	ST8000DM002	8TB	9,354	80.6	842,239	114	4.94%
Seagate	ST8000NM000A	8TB	153	11.6	12,088	-	0.00%
Seagate	ST8000NM0055	8TB	14,118	68.8	1,270,271	215	6.18%
Seagate	ST10000NM0086	10TB	1,124	66.4	100,772	34	12.31%
Seagate	ST12000NM0007	12TB	1,214	43.6	109,092	25	8.36%
Seagate	ST12000NM0008	12TB	19,677	38.8	1,763,868	157	3.25%
Seagate	ST12000NM001G	12TB	13,029	29.8	1,157,666	44	1.39%
Seagate	ST14000NM0018	14TB	60	14.1	5,111	2	14.28%
Seagate	ST14000NM001G	14TB	10,790	28.5	968,724	52	1.96%
Seagate	ST14000NM0138	14TB	1,458	30.8	131,819	37	10.25%
Seagate	ST16000NM001G	16TB	27,255	15.3	2,242,685	54	0.88%
Seagate	ST16000NM002J	16TB	309	12.5	27,513	-	0.00%
Toshiba	MD04ABA400V	4TB	94	97.3	8,366	-	0.00%
Toshiba	HDWF180	8TB	61	19.2	5,577	3	19.63%
Toshiba	MG07ACA14TA	14TB	38,101	31.8	3,426,456	133	1.42%
Toshiba	MG07ACA14TEY	14TB	616	24.1	52,749	1	0.69%
Toshiba	MG08ACA16TA	16TB	5,199	13.6	467,288	4	0.31%
Toshiba	MG08ACA16TE	16TB	5,923	20.6	527,557	21	1.45%
Toshiba	MG08ACA16TEY	16TB	5,289	18.8	470,668	-	0.00%
WDC	WUH721414ALE6L4	14TB	8,432	30.6	759,062	16	0.77%
WDC	WUH721816ALE6L0	16TB	2,697	20.8	239,957	-	0.00%
WDC	WUH721816ALE6L4	16TB	14,099	9.3	1,256,975	13	0.38%
		Totals	240,940		21,408,175	1,339	2.28%

- BackBlaze: Consultora dedicada al almacenamiento en la nube
  - Cada trimestre publica un informe detallando:
    - Ratio de fallos en sus discos duros
    - Comparativas de rendimiento
    - Datos históricos

Comando rsync

- Herramienta GNU para backups
- Forma de uso más simple:
  - rsync [opciones] <origen> <destino>
- Opciones:
  - -v
  - -a Mantiene usuarios
  - -z Comprime antes de copiar
  - -h Mostrar tasas de transferencia y tamaños (MB/s en vez de bytes/s)
- Ejemplo: **rsync -vazh /home /dev/sdc**
- Se suele utilizar para copias remotas por red

Comando rsnapshot

- Herramienta basada en rsync para realizar copias incrementales, gestionando un histórico de las mismas con rotación
- No viene instalada en Ubuntu Server por defecto
  - Instalar con “apt install rsnapshot”
- Configuración: /etc/rsnapshot.conf
- Uso:
  - **rsnapshot configtest** Verifica que la configuración es correcta
  - **rsnapshot <TAG>** Realiza una copia del tipo <TAG>, p.e. “daily”
  - **rsnapshot-diff** Compara 2 copias hechas en instantes diferentes

Alternativas:

- Rudimentarias:
  - Comando tar → Combinándolo con herramientas de compresión (bzip, zip)
  - Comando dd → dd if=/dev/sda2 of=/dev/tape
  - Comando cp -a → Para replicar contenido de disco a nivel de fichero
- Comerciales:
  - HP Data Protector
  - IBM Spectrum Protect (Tivoli Storage Manager)