Engineer

RAID

☰ Varios discos operan como uno y suelen tolerar que falle uno

Qué es

RAID hace funcionar varios discos como un único dispositivo de bloques para que el conjunto pueda ser más rápido, tener más capacidad o resistir mejor los fallos que cualquiera de sus miembros. El nombre procede de un informe técnico de 1987 de la Universidad de California en Berkeley, de David Patterson, Garth Gibson y Randy Katz: *A Case for Redundant Arrays of Inexpensive Disks (RAID)*, sobre el uso de conjuntos redundantes de discos económicos. Linux escribe los niveles en minúsculas y sin espacio, raid1 o raid5, tal como los comunica su controlador md.

El informe parte de un desajuste. Los procesadores y la memoria se hacían más rápidos, y el resumen advierte de que esa mejora se desperdicia si la entrada y la salida no siguen el ritmo. Compara el Single Large Expensive Disk, o SLED, un único disco grande y caro cuya capacidad había crecido rápidamente mientras su rendimiento solo había mejorado de forma modesta, con conjuntos de discos baratos fabricados para ordenadores personales, de los que espera mejoras de un factor aproximado de diez frente al disco grande en velocidad, fiabilidad, consumo eléctrico y posibilidades de crecimiento. La comparación es concreta: un disco IBM 3380 para grandes sistemas, un Fujitsu Super Eagle para miniordenadores y el Conner CP 3100 de 100 MB para ordenadores personales.

Después llega el inconveniente, en una sección que los autores presentan como malas noticias. Si los discos fallan de forma independiente y a una tasa constante, que según el informe es lo que suponen los fabricantes al indicar un tiempo medio hasta el fallo (MTTF), el MTTF de un conjunto es el de un disco dividido por el número de discos. Cien CP 3100, cada uno con una especificación de 30.000 horas, dan 300 horas, menos de dos semanas; mil dan 30 horas. La conclusión es tajante: «Sin tolerancia a fallos, los grandes conjuntos de discos económicos son demasiado poco fiables para resultar útiles».

La respuesta es la redundancia. Los discos se dividen en grupos de fiabilidad, cada uno con discos de comprobación adicionales que contienen información redundante. Cuando falla un disco, se sustituye y su contenido se reconstruye en el nuevo a partir de esa información; el tiempo que tarda es el tiempo medio de reparación, y los discos de reserva en caliente, que entran en servicio electrónicamente cuando muere un disco, lo acortan. El informe expone después cinco formas de organización y denomina nivel a cada una. El nivel 1 replica cada disco en espejo. El nivel 2 corrige errores con un código de Hamming, como hacían los sistemas de memoria. El nivel 3 solo necesita un disco de comprobación por grupo. El nivel 4 permite que pequeñas lecturas y escrituras se dirijan a discos separados al mismo tiempo. El nivel 5 distribuye los datos y la información de comprobación entre todos los discos, de modo que ningún disco de comprobación por sí solo frene las escrituras.

La práctica ha ampliado la lista. mdadm, la herramienta que gestiona RAID por software en Linux, acepta linear, raid0 (o stripe), raid1, raid4, raid5, raid6 y raid10, entre otros. La distribución en bandas sin redundancia, hoy llamada nivel 0, no figura en la clasificación de 1987, que empieza por los discos en espejo. El informe dejó abierta la elección entre construir un conjunto por hardware o por software y la consideró una cuestión exclusivamente de costes y beneficios. Linux md es del tipo por software; una tarjeta controladora que presenta al sistema operativo un único disco grande es del tipo por hardware.

Utilice el concepto en una decisión

Aplique «¿Qué es RAID?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Copias de seguridad que se prueban en vez de darse por buenas y un plan escrito para la mañana en que un servidor no vuelva a encenderse.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 100 copias críticas automatizadas con Barman — 100 copias críticas automatizadas en las bases de datos

100 copias críticas automatizadas con Barman Póngase en contacto →

Por qué importa

Un disco se desgasta y, en un servidor con un solo disco, su fallo detiene la máquina y deja los datos a merced de las copias de seguridad que existan. El informe abre su sección sobre fiabilidad precisamente con eso: los discos son lo bastante poco fiables como para que los administradores de sistemas copien todo cada noche. Con redundancia, un disco averiado se convierte en una

reparación en lugar de una interrupción. El conjunto sigue funcionando con los miembros restantes, en un estado que Linux denomina degraded (degradado), y recupera su capacidad completa una vez añadido y reconstruido el sustituto.

Esa protección solo dura mientras aguanten los discos restantes. La aritmética del informe considera que se pierden datos cuando falla un segundo disco del mismo grupo antes de haber reparado el primero, por lo que la duración de la ventana de reparación importa tanto como el número de discos de comprobación. Con reservas en caliente y reparaciones cortas, las cifras resultan impresionantes: el informe calcula que un conjunto de nivel 5 de 1.000 discos, en grupos de diez y con un 5 por ciento de reservas en espera, seguiría teniendo un MTTF de 90 años. En otra parte advierte de que un terremoto o una sobretensión pueden hacer que los discos de un mismo conjunto fallen juntos, y sus fórmulas suponen que no ocurre.

Elegir un nivel exige equilibrar capacidad, velocidad y coste. La conclusión del propio informe es que los discos en espejo sirven para el procesamiento de transacciones que no utiliza más de la mitad del almacenamiento, y que el nivel 5 parece mejor cuando se necesita más capacidad, para trabajo de supercomputación o para una mezcla de este y el procesamiento de transacciones.

Cuando se agota la redundancia, queda la recuperación. La entrada del portafolio sobre recuperación de datos de tarjetas SD, discos duros, SSD y RAID trata un conjunto RAID averiado como un problema propio, distinto de un SSD muerto o de un disco de platos averiado, con un enfoque adecuado que depende del medio.

Cómo se aplica en la práctica

Tomemos un servidor con una partición en cada uno de dos discos, `/dev/sda1` y `/dev/sdb1`, que se van a replicar en espejo para un volumen de datos. Con Linux md, tres comandos construyen el conjunto, le añaden un sistema de archivos y lo registran en la configuración:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
mke2fs -t ext4 -L data /dev/md0
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

El primero crea `/dev/md0`, un conjunto RAID1 de las dos particiones, y el núcleo inicia una resincronización que recalcula la redundancia, como después de cualquier creación. El segundo formatea el conjunto exactamente como un disco individual; el sistema de archivos ve un solo dispositivo. El tercero añade una línea que describe el conjunto al archivo de configuración de mdadm, situado en `/etc/mdadm.conf` o `/etc/mdadm/mdadm.conf`, y el manual pide revisar esa salida antes de usarla, porque puede contener detalles que no se quieran incluir.

Tres archivos muestran el estado del conjunto:

```
cat /proc/mdstat
cat /sys/block/md0/md/degraded
cat /sys/block/md0/md/sync_action
```

`/proc/mdstat` enumera cada dispositivo md activo con su estado, y mdadm lo lee para encontrar conjuntos y seguir su reconstrucción. Cada conjunto también tiene un directorio bajo `/sys/block/`, donde `degraded` contiene el número de miembros que faltan: 0 para un conjunto completo y 1 si falta un disco. `sync_action` indica `idle` cuando no ocurre nada, `resync` después de la creación o de un apagado incorrecto, `recover` mientras una reserva en caliente asume el contenido de un miembro perdido, y `check` o `repair` durante una verificación completa. Escribir `check` en él inicia esa verificación, que lee todos los bloques del conjunto:

```
echo check > /sys/block/md0/md/sync_action
cat /sys/block/md0/md/mismatch_cnt
```

Después, `mismatch_cnt` indica el número de sectores que se reescribieron o que se habrían reescrito. La documentación del núcleo señala que la mayoría de los niveles trabajan con páginas en lugar de sectores, por lo que el recuento puede superar el número de errores reales.

Alguien también tiene que enterarse de un fallo. `mdadm --monitor` consulta periódicamente los conjuntos y convierte lo que detecta en eventos, que puede enviar por correo a una dirección o entregar a un programa. En un sistema gestionado por systemd, el manual cita `mdmonitor.service` como vía principal y

`MAILADDR` en `mdadm.conf` como ajuste que debe rellenarse para el correo. Solo cinco eventos envían correo: `Fail` , `FailSpare` , `DegradedArray` , `SparesMissing` y `TestMessage` . El último existe para demostrar que la vía funciona:

```
MAILADDR ops@example.com
```

```
mdadm --monitor --scan --oneshot --test
mdadm --detail --test /dev/md0
```

El primero envía un mensaje de prueba por cada conjunto que encuentra y después termina. El segundo sirve para un script: con `--test` , su código de salida es 0 para un conjunto sano, 1 cuando ha fallado al menos un dispositivo, 2 cuando han fallado tantos que el conjunto resulta inutilizable y 4 cuando mdadm encuentra un error al recopilar información sobre el conjunto.

El conjunto cubre un suceso, la muerte de un disco, y nada más. La entrada del portafolio sobre automatización de copias de seguridad y pruebas de restauración describe un diseño de recuperación separado: instantáneas restic cifradas y deduplicadas, enviadas por SFTP a almacenamiento de otro proveedor, temporizadores systemd y un flujo de trabajo mensual que restaura un archivo conocido en una ubicación temporal y compara sus bytes. Se trata de una prueba de restauración prevista en el diseño, no de una demostración de que se haya ejecutado. El registro de implementación fechado el 2026-10-27 indica que los temporizadores estaban deshabilitados al no haber un repositorio configurado y no registra ninguna restauración real completada.

Dónde surgen los problemas

Un conjunto está diseñado para un suceso, la muerte de uno de sus discos, y la mayoría de las formas de perder datos quedan justo fuera de ese suceso.

El conjunto se confunde con una copia de seguridad. Un espejo escribe cada cambio en ambos discos a la vez, incluido un archivo borrado por error, una tabla eliminada por una migración defectuosa o un volumen cifrado por ransomware. Todos los miembros contienen entonces el mismo daño. La solución es una copia de seguridad separada, conservada fuera del host, con restauraciones probadas según una programación, de modo que el conjunto proteja frente a fallos del hardware y la copia frente a todo lo demás.

Un conjunto degradado pasa inadvertido. Un conjunto degradado sigue atendiendo peticiones y nada parece distinto hasta que falla el siguiente disco. mdadm no envía correo salvo que tenga una dirección o un programa: con `--scan` y sin ninguno de los dos configurado, el manual dice que no monitoriza nada. `DegradedArray` también cubre únicamente los conjuntos que ya están degradados cuando mdadm los ve por primera vez; un disco que muere después produce un evento `Fail`. La solución es configurar `MAILADDR`, comprobar que `mdmonitor.service` está en ejecución y enviar un mensaje `--test` para demostrar que llega el correo.

El segundo disco falla durante la reconstrucción. Una reconstrucción recupera el disco que falta a partir de todos los demás, y el modelo del informe pierde datos exactamente cuando falla otro disco del grupo dentro de ese tiempo de reparación. Dejar un disco averiado sin sustituir durante una semana amplía la ventana a una semana. La solución es acortarla: mantenga una reserva en caliente en el conjunto, que mdadm puede configurar al crearlo con `--spare-devices`, sustituya pronto los discos averiados y ejecute un `check` programado para detectar un disco con problemas mientras el conjunto aún puede corregirlos.

Los discos no son independientes. La aritmética del MTTF supone que los discos fallan por separado, y el propio informe cita un terremoto y una sobretensión como casos en los que no sería así. Dos discos en una carcasa comparten alimentación eléctrica, controladora, sala y edificio. La solución es tratar la cifra calculada como un límite superior y cubrir las causas compartidas con algo situado fuera de la caja, como una copia en otro emplazamiento.

Un conjunto RAID 5 o 6 está sucio y degradado a la vez. Si la máquina se detiene incorrectamente mientras falta un miembro, no se puede confiar en la paridad y algunos bloques no pueden reconstruirse, por lo que la documentación del núcleo advierte de corrupción indetectable y md se niega a iniciar el conjunto. Forzarlo con `mdadm --assemble --force`, o con el parámetro de módulo `start_dirty_degraded` para un sistema de archivos raíz, supone aceptar ese riesgo. La solución es elegir de antemano una política de consistencia: md puede registrar las escrituras en un dispositivo de diario para los niveles 4, 5 y 6, y para el nivel 5 ofrece un registro de paridad parcial que, según la documentación, cierra el agujero de escritura. Cuando forzar el inicio sea la única forma de acceder, considere sospechosos los datos y compárelos con una copia de seguridad.

RAID 0 se cuenta como protección. md indica que los conjuntos raid0 y linear no contienen información de redundancia, y el monitor de mdadm los señala con `Wrong-Level` porque no admiten ninguna de las operaciones de fallo, reserva y resincronización que vigila. No hay nada con lo que reconstruir un miembro que falte. La solución es usar el nivel 0 solo para datos que puedan desecharse o recrearse, y no contarlo nunca como RAID en el sentido del informe de 1987.

Una controladora de hardware oculta los discos. Detrás de una tarjeta controladora, el sistema operativo ve un dispositivo lógico, y preguntar a ese dispositivo por la salud de los discos llega a la controladora, no a los discos. smartctl, por ejemplo, necesita `-d megaraid,N` para dirigirse al disco N detrás de una controladora MegaRAID; su manual indica que esa opción también sirve para las tarjetas Dell PERC. La solución es monitorizar cada disco físico a través de la interfaz de la propia controladora para detectar un miembro que falla mientras el conjunto sigue completo.

Lecturas adicionales

- A Case for Redundant Arrays of Inexpensive Disks (RAID) — La página del informe técnico de Berkeley, con su resumen y el PDF.
- UCB/CSD-87-391, informe completo (PDF) — La aritmética del MTTF, los grupos de fiabilidad y los cinco niveles.
- Conjuntos RAID, documentación del núcleo Linux — Los archivos de md bajo `/sys`, los conjuntos sucios y degradados y las políticas de consistencia.
- mdadm(8) — Creación de conjuntos, eventos del monitor, correo y códigos de salida de `--detail --test`.
- smartctl(8), smartmontools — Acceso a discos detrás de controladoras RAID mediante la opción `-d`.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es RAID?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/raid/