RAID-Software CÓMO

Jakob Østergaard, *jakob@ostenfeld.dk*
traducido por Juan Piernas Cánovas, *piernas@ditec.um.es*
v. 0.90.3 - Alpha, 22 mayo 1999

Este CÓMO describe cómo usar un RAID software bajo Linux. Debería usar los parches RAID disponibles en *ftp://ftp.fi.kernel.org/pub/linux/daemons/raid/alpha*. El CÓMO original en inglés se puede encontrar en *http://ostenfeld.dk/~jakob/Software-RAID.HOWTO/*.

Índice General

1 Introducción

Este CÓMO ha sido escrito por Jakob Østergaard basándose en un gran número de mensajes de correo entre el autor, Ingo Molnar (*mingo@chiara.csoma.elte.hu*) – uno de los desarrolladores de RAID –, la lista de correo linux-raid (*linux-raid@vger.rutgers.edu*) y diversas personas.

La razón por la que se ha escrito este CÓMO, a pesar de existir ya un RAID-Software CÓMO es que el anterior describe el estilo antiguo de RAID por software, presente en los núcleos existentes. Este CÓMO describe el uso del «nuevo estilo» de RAID que se ha desarrollado más recientemente. Éste nuevo estilo de RAID tiene muchas características no presentes en el anterior.

Parte de la información de este CÓMO le puede parecer trivial si es un entendido en RAID. Sáltese esas partes.

1.1 Renuncia de responsabilidad

La declinación obligatoria de responsabilidades:

Aunque el código RAID tratado aquí me ha parecido ser estable, y para muchas otras personas, puede no funcionar para Usted. Si pierde todos sus datos, su trabajo, es golpeado por un camión o cualquier otra cosa, no será culpa mía ni de los desarrolladores. ¡Conciéncese de que usa el software RAID y esta información por su cuenta y riesgo!. No hay ningún tipo de garantía de que ningún software ni esta información sean correctos en modo alguno, ni adecuados para cualquier tipo de uso en particular. Haga copia de seguridad de sus datos antes de experimentar con esto. Más vale prevenir que lamentarse.

1.2 Requisitos

Este CÓMO asume que está usando alguna de las últimas versiones `2.2.x` o `2.0.x` del núcleo con un parche `raid0145` acorde y la versión `0.90` del paquete `raidtools`. Ambos se pueden encontrar en *ftp://ftp.fi.kernel.org/pub/linux/daemons/raid/alpha*. El parche RAID, el paquete `raidtools` y el núcleo deben concordar tanto como sea posible. En ocasiones puede ser necesario usar un núcleo antiguo si no hay parches raid disponibles para el último.

2 ¿Por qué RAID?

Puede haber muchas buenas razones para usar RAID. Unas pocas son: la posibilidad de combinar varios discos físicos en un único dispositivo «virtual» más grande, o mejoras en el rendimiento y redundancia.

2.1 Detalles técnicos

El RAID de Linux puede funcionar sobre la mayoría de los dispositivos de bloque. No importa si usa dispositivos IDE, SCSI o una mezcla de ambos. Incluso algunas personas han usado dispositivo de bloque en red (Network Block Device, NBD) con diferentes grados de éxito.

Asegúrese de que el bus (o buses) de los discos son lo suficientemente rápidos. No debería tener 14 discos UW-SCSI en un único bus UW, si cada disco puede dar 10MB/s y el bus sólo puede sostener 40MB/s. Además, sólo debería tener un dispositivo por bus IDE. El uso de discos como maestro/esclavo es funesto para el rendimiento. IDE es realmente ineficiente accediendo a más de un disco por bus. Naturalmente, todas las placas madre modernas tienen dos buses IDE, por lo que puede configurar dos discos en RAID sin comprar más tarjetas controladoras.

La capa RAID no tiene absolutamente nada que ver con la capa del sistema de ficheros. Puede poner cualquier sistema de ficheros sobre un dispositivo RAID, tal y como haría con cualquier otro dispositivo de bloques.

2.2 Términos

La palabra *RAID* se refiere a *RAID por software de Linux*. Este CÓMO no trata ningún aspecto de RAID por hardware.

Cuando se describen configuraciones, es útil referirse al número de discos y sus tamaños. En todos los casos se usa la letra N para denotar el número de discos activos en el array (sin contar los discos de reserva). La letra S es el tamaño del disco más pequeño en el array, a menos que se diga otra cosa. La letra P representa el rendimiento de un disco en el array, en MB/s. Cuando se use, supondremos que los discos son igual de rápidos, lo cual no siempre puede ser cierto.

Note que se asume que las palabras *dispositivo* y *disco* significan lo mismo. Normalmente, los dispositivos usados para construir un dispositivo RAID son particiones de discos, no necesariamente discos enteros. Pero, normalmente, combinar varias particiones de un mismo disco no tiene sentido, por lo que las palabras *dispositivo* y *disco* simplemente significan *particiones de discos diferentes*.

2.3 Niveles RAID

Lo siguiente es una breve descripción de lo que soportan los parches RAID de Linux. Parte de esta información es información RAID absolutamente básica, aunque he añadido unas pocas reseñas de lo que hay de especial en la implementación de Linux de los niveles. Simplemente, sáltese esta sección si conoce RAID. Regrese después cuando tenga problemas :)

Los actuales parches RAID para Linux soportan los siguientes niveles:

- **Modo Lineal (Linear mode)**
  - Dos o más discos se combinan en un único dispositivo físico. Los discos se «adjuntan» unos a otros de tal manera que las escrituras en el dispositivo RAID primero llenarán el disco 0, a continuación el disco 1 y así sucesivamente. Los discos no tienen porqué ser del mismo tamaño. De hecho, los tamaños no importan para nada aquí :)
  - No existe redundancia en este nivel. Si un disco falla perderá toda su información con toda probabilidad. Sin embargo, puede tener suerte y recuperar algunos datos, ya que el sistema de ficheros simplemente habrá perdido un gran puñado de datos consecutivos.
  - El rendimiento de las lecturas y las escrituras no se incrementará para lecturas/escrituras individuales. Pero si varios usuarios usan el dispositivo, puede tener la suerte de que un usuario use efectivamente el primer disco y el otro usuario acceda a ficheros que por casualidad residan en el segundo disco. Si esto ocurre, verá un aumento en el rendimiento.
- **RAID-0**
  - También llamado modo *striping* o de distribución por bandas. Como el modo lineal salvo que las lecturas y escrituras se realizan en paralelo en los dispositivos. Éstos deben tener aproximadamente el mismo tamaño. Puesto que todos los accesos se realizan en paralelo, los discos se llenan por igual. Si un dispositivo es mucho mayor que los otros demás, el espacio extra se utilizará en el dispositivo RAID durante las escrituras en el extremo superior, aunque sólo se accederá a este disco más grande. Naturalmente, esto perjudica el rendimiento.
  - Como en el modo lineal, tampoco hay redundancia en este nivel. A diferencia del modo lineal, no será capaz de recuperar ningún dato si un disco falla. Si elimina un disco de un grupo RAID-0, el dispositivo RAID no perderá simplemente un bloque consecutivo de datos, sino que se llenará con pequeños agujeros por todo el dispositivo. Probablemente, `e2fsck` no sea capaz de recuperar gran cosa.
  - El rendimiento de las lecturas y las escrituras se incrementará, ya que las lecturas y las escrituras se realizan en paralelo sobre los dispositivos. Normalmente, ésta es la razón principal para usar RAID-0. Si los buses a los discos son suficientemente rápidos, puede obtener casi *N*P* MB/seg.
- **RAID-1**
  - Este es el primer modo que realmente tiene redundancia. RAID-1 se puede usar en dos o más discos con cero o más discos de reserva. Este modo mantiene en un disco un duplicado exacto de la información del otro(s) disco(s). Por supuesto, los discos deben ser del mismo tamaño. Si un disco es mayor que otro, su dispositivo RAID será del tamaño del disco más pequeño.
  - Si se eliminan (o fallan) hasta *N-1* discos, los datos permanecerán intactos. Si existen discos de reserva disponibles y el sistema (es decir, las controladoras SCSI o los chipsets IDE, etc.) sobreviven al desastre, comenzará inmediatamente la reconstrucción de un duplicado en uno de los discos de reserva, después de la detección del fallo del disco.

  - Normalmente, el rendimiento de las lecturas aumenta hasta casi *N*P*, mientras que el rendimiento de las escrituras es el mismo que el de un único dispositivo o, tal vez, incluso menos. Las lecturas se pueden hacer en paralelo pero, cuando se escribe, la CPU debe transferir N veces la cantidad de datos que normalmente transferiría (recuerde, se deben enviar N copias idénticas de todos los datos a los discos).

- **RAID-4**
  - Este nivel de RAID no se usa con mucha frecuencia. Se puede usar sobre 3 o más discos. En lugar de duplicar completamente la información, guarda información de paridad en un único disco y escribe datos a los otros discos de forma parecida a un RAID-0. Ya que uno de los discos se reserva para información de paridad, el tamaño del array será *(N-1)*S*, donde S es el tamaño del disco más pequeño del array. Como en un RAID-1, los discos deben ser del mismo tamaño, o de lo contrario tendrá que aceptar que el valor de S en la fórmula *(N-1)*S* anterior será el tamaño del disco más pequeño del array.
  - Si un disco falla, **y no es el de paridad**, se puede usar la información de paridad para reconstruir todos los datos. Si dos discos fallan, se perderá toda la información. .
  - La razón por la que este nivel no se usa con mucha frecuencia es que la información de paridad se guarda en un único disco. Esta información se debe actualizar *cada* vez que se escribe en uno de los otros discos. Por eso, el disco de paridad se convertirá en un cuello de botella si no es mucho más rápido que los otros discos. Sin embargo, si por pura casualidad tuviera muchos discos lentos y un disco muy rápido, este nivel de RAID podría resultarle muy útil.

- **RAID-5**
  - Este es quizás el modo RAID más útil cuando uno desea combinar un mayor número de discos físicos y todavía conservar alguna redundancia. RAID-5 se puede usar sobre 3 o más discos, con cero o más discos de reserva. El tamaño del dispositivo RAID-5 resultante será *(N-1)*S*, tal y como sucede con RAID-4. La gran diferencia entre RAID-5 y RAID-4 es que la información de paridad se distribuye uniformemente entre los discos participantes, evitando el problema del cuello de botella del RAID-4.
  - Si uno de los discos falla, todos los datos permanecerán intactos, gracias a la información de paridad. Si existen discos de reserva disponibles, la reconstrucción comenzará inmediatamente después del fallo del dispositivo. Si dos discos fallan simultáneamente, todos los datos se perderán. RAID-5 puede sobrevivir a un fallo de disco, pero no a dos o más.
  - Normalmente, el rendimiento de las lecturas y las escrituras se incrementará, pero es difícil predecir en qué medida.

2.3.1 Discos de reserva

Los discos de reserva son discos que no forman parte del grupo RAID hasta que uno de los discos activos falla. Cuando se detecta un fallo de disco, el dispositivo se marca como *defectuoso* y la reconstrucción se inicia inmediatamente sobre el primer disco de reserva disponible.

De esta manera, los discos de reserva proporcionan una buena seguridad extra, especialmente a sistemas RAID-5 que tal vez, sean difíciles de lograr (físicamente). Se puede permitir que el sistema funcione durante algún tiempo con un dispositivo defectuoso, ya que se conserva toda la redundancia mediante los discos de reserva.

No puede estar seguro de que su sistema sobrevivirá a una caída de disco. La capa RAID puede que maneje los fallos de dispositivos verdaderamente bien, pero las controladoras SCSI podrían fallar durante el manejo del error o el chipset IDE podría bloquearse, o muchas otras cosas.

2.4 Espacio de intercambio (*swap*) sobre RAID

No hay ninguna razón para usar RAID a fin de aumentar el rendimiento del sistema de paginación de memoria (*swap*). El propio núcleo puede balancear el intercambio entre varios dispositivos si simplemente les da la misma prioridad en el fichero `/etc/fstab`.

Un buen fstab se parece a éste:

```
/dev/sda2       swap           swap    defaults,pri=1   0 0
/dev/sdb2       swap           swap    defaults,pri=1   0 0
/dev/sdc2       swap           swap    defaults,pri=1   0 0
/dev/sdd2       swap           swap    defaults,pri=1   0 0
/dev/sde2       swap           swap    defaults,pri=1   0 0
/dev/sdf2       swap           swap    defaults,pri=1   0 0
/dev/sdg2       swap           swap    defaults,pri=1   0 0
```

Esta configuración permite a la máquina paginar en paralelo sobre siete dispositivos SCSI. No necesita RAID, ya que esa ha sido una característica del núcleo desde hace mucho tiempo.

Otra razón por la que podría interesar usar RAID para *swap* es la alta disponibilidad. Si configura un sistema para arrancar desde, por ejemplo, un dispositivo RAID-1, el sistema podría ser capaz de sobrevivir a un fallo de disco. Pero si el sistema ha estado paginando sobre el ahora dispositivo defectuoso, puede estar seguro de que se vendrá abajo. El intercambio sobre un dispositivo RAID-1 solucionaría este problema.

Sin embargo, el intercambio sobre RAID-{1,4,5} **NO** está soportado. Puede configurarlo, pero fracasará. La razón es que la capa RAID algunas veces reserva memoria antes de realizar una escritura. Esto produce un bloqueo, quedando en un punto muerto, ya que el núcleo tendrá que reservar memoria antes de que pueda intercambiar, e intercambiar antes de que pueda reservar memoria.

Es triste pero cierto, al menos por ahora.

3 Asuntos hardware

Esta sección mencionará algunos de los asuntos hardware involucrados en el funcionamiento de un RAID software.

3.1 Configuración IDE

En efecto, es posible hacer funcionar un RAID sobre discos IDE. También se puede obtener un rendimiento excelente. De hecho, el precio actual de los discos y las controladoras IDE hacen de IDE algo a tener en cuenta cuando se montan nuevos sistemas RAID.

- **Estabilidad física:** tradicionalmente, los discos IDE han sido de peor calidad mecánica que los discos SCSI. Incluso hoy en día, la garantía de los discos IDE es típicamente de un año, mientras que, a menudo, es de 3 a 5 años en los discos SCSI. Aunque no es justo decir que los discos IDE son por definición de menor calidad, uno debería ser consciente de que los discos IDE de *algunas* marcas *pueden* fallar con más frecuencia que los discos SCSI similares. Sin embargo, otras marcas usan exactamente la misma estructura mecánica tanto para los discos SCSI como para los discos IDE. Todo se reduce a: todos los discos fallan, tarde o temprano, y uno debería estar preparado para ello.
- **Integridad de los datos:** al principio, IDE no tenía forma de asegurar que los datos enviados a través del bus IDE eran los mismos que los datos escritos realmente en el disco. Esto se debió a la falta total de paridad, sumas de verificación (*checksums*), etc. Ahora, con el estándar *UltraDMA*, los dispositivos

IDE realizan una suma de verificación sobre los datos que reciben y por eso es altamente improbable que los datos se corrompan.

- **Rendimiento:** no voy a escribir aquí sobre el rendimiento de IDE de forma detallada. Una historia realmente breve sería:
  - Los dispositivos IDE son rápidos (12 MB/s y más)
  - IDE tiene una mayor sobrecarga de CPU que SCSI (pero, ¿a quién le preocupa?)
  - Sólo usa **un** disco IDE por bus, los discos esclavos deterioran el rendimiento.
- **Resistencia a los fallos:** la controladora IDE normalmente sobrevive a un dispositivo IDE que ha fallado. La capa RAID marcará el disco como defectuoso y, si está trabajando con un RAID de nivel 1 o superior, la máquina debería trabajar igual de bien hasta que la desconecte para su mantenimiento.

Es **muy** importante que sólo use **un** disco IDE por bus IDE. Dos discos no sólo arruinarían el rendimiento sino que, también, el fallo de un disco a menudo garantiza el fallo del bus y, por tanto, el fallo de todos los discos de ese bus. En una configuración RAID tolerante a fallos (RAID de niveles 1, 4, 5) el fallo de un disco se puede manejar, pero el fallo de dos discos (los dos discos del bus que ha fallado debido a uno de ellos) dejará el array inutilizable. También, el dispositivo esclavo o la controladora IDE de un bus pueden confundirse de manera horrible cuando el dispositivo maestro del bus falla. Un bus, un disco, esa es la regla.

Existen controladoras IDE PCI baratas. A menudo puede obtener 2 o 4 buses por unos 80 dólares. Considerando el precio mucho más bajo de los discos IDE respecto a los discos SCSI, diría que un array de discos IDE podría ser una solución realmente buena si uno puede vivir con los relativamente pocos discos (unos 8 probablemente) que se pueden conectar a un sistema típico (a menos que, naturalmente, tenga muchas ranuras PCI para dichas controladoras IDE).

3.2 Cambio de discos en caliente (*Hot-Swap*)

Éste ha sido un tema de actualidad en la lista linux-kernel durante algún tiempo. Aunque el intercambio en caliente de los dispositivos está soportado hasta cierto punto, todavía no es algo que se pueda hacer fácilmente.

3.2.1 Intercambio en caliente de dispositivos IDE

¡No lo haga! IDE no soporta en modo alguno el intercambio en caliente. Seguro, puede funcionar para usted si compila el soporte IDE como módulo (sólo posible en la serie `2.2.x` del núcleo) y lo vuelve a cargar después de que haya reemplazado el dispositivo. Pero también puede terminar perfectamente con una controladora IDE frita y observará que el período de dicho sistema fuera de servicio será mucho mayor que habiendo reemplazado el dispositivo con el sistema apagado.

El principal problema, aparte de los aspectos eléctricos que pueden destruir su hardware, es que se debe reexplorar el bus IDE después de que se hayan intercambiado los discos. El manejador IDE actual no puede hacer eso. Si el nuevo disco es 100% idéntico al antiguo (geometría, etc.) *puede* que funcione incluso sin volver a explorar el bus pero, créame, aquí está caminando por *el filo de la navaja*.

3.2.2 Intercambio en caliente (*Hot-Swap*) de dispositivos SCSI

El hardware SCSI normal tampoco es capaz de soportar intercambios en caliente. Sin embargo, *puede* que funcione. Si su manejador SCSI soporta la reexploración del bus y la conexión y desconexión de dispositivos, puede ser capaz de intercambiar dispositivos en caliente. Sin embargo, en un bus SCSI normal probablemente no debería desenchufar dispositivos mientras su sistema esté todavía encendido. Pero, le repito, puede que funcione simplemente (y también puede terminar con su hardware frito).

La capa SCSI **debería** sobrevivir si un disco muere, pero no todos los manejadores SCSI soportan esto todavía. Si su manejador SCSI muere cuando un disco cae, su sistema caerá con él y la conexión en caliente no será verdaderamente interesante entonces.

3.2.3 Intercambio en caliente con SCA

Con SCA debería ser posible conectar dispositivos en caliente. Sin embargo, no poseo el hardware para probar esto y no he oído de nadie que lo haya probado, por lo que verdaderamente no puedo dar ninguna receta de cómo hacer esto.

De todos modos, si quiere jugar con esto, debería conocer los aspectos internos de SCSI y de RAID. Por tanto, no voy a escribir aquí nada que no pueda comprobar que funciona. En cambio, sí puedo proporcionarle algunas pistas:

- Busque la cadena `remove-single-device` en `linux/drivers/scsi/scsi.c`
- Eche un vistazo a `raidhotremove` y `raidhotadd`

No todos los manejadores SCSI soportan la conexión y desconexión de dispositivos. En la serie 2.2 del núcleo, al menos los manejadores de la controladoras Adaptec 2940 y Symbios NCR53c8xx parecen soportarlo, mientras que otras puede que sí o puede que no. Agradecería que alguien me pasara más información sobre esto...

4 Configuración de RAID

4.1 Configuración general

Esto es lo que necesita para cualquiera de los niveles RAID:

- Un núcleo. Obtenga la versión `2.0.36` o un núcleo `2.2.x` reciente.
- Los parches RAID. Normalmente existe un parche disponible para los núcleos recientes.
- El paquete de herramientas RAID (`raidtools`).
- Paciencia, una pizza y su bebida con cafeína favorita.

Todo este software se puede encontrar en `ftp://ftp.fi.kernel.org/pub/linux`;. Las herramientas RAID y los parches están en el subdirectorio `daemons/raid/alpha`. Los núcleos se encuentran en el subdirectorio `kernel`.

Parchee el núcleo, configúrelo para incluir el soporte del nivel RAID que quiera usar. Compílelo e instálelo.

A continuación desempaquete, configure, compile e instale las herramientas RAID.

Hasta ahora todo va bien. Si rearranca ahora, debería tener el fichero `/proc/mdstat`. Recuérdelo, ese fichero es su amigo. Vea lo que contiene haciendo `cat /proc/mdstat`. Le debe decir que tiene registrada la personalidad RAID (es decir, el modo RAID) correcta y que actualmente no hay dispositivos RAID activos.

Cree las particiones que quiere incluir en su grupo RAID.

Ahora, vayamos a un modo específico.

4.2 Modo lineal

De acuerdo, así que tiene dos o más particiones que no son necesariamente del mismo tamaño (pero que, naturalmente, pueden serlo) que quiere adjuntar unas con otras.

Prepare el fichero `/etc/raidtab` para describir su configuración. He preparado un `/etc/raidtab` para dos discos en modo lineal y el fichero se parece a esto:

```
raiddev /dev/md0
        raid-level      linear
        nr-raid-disks   2
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Aquí no se soportan discos de reserva. Si un disco muere, el array muere con él. No hay información que poner en un disco de reserva.

Creemos el array. Ejecute la orden:

```
mkraid /dev/md0
```

Esto inicializará su array, escribirá superbloques *persistentes* y arrancará el array.

Échele un vistazo a `/proc/mdstat`. Debe ver que el array está funcionando.

Ahora, puede crear un sistema de ficheros, justo como haría con cualquier otro dispositivo, montarlo, incluirlo en su `/etc/fstab`, etc.

4.3 RAID-0

Tiene dos o más dispositivos, de aproximadamente el mismo tamaño, y quiere combinar sus capacidades de almacenamiento y rendimiento accediéndolos en paralelo.

Prepare el fichero `/etc/raidtab` para describir su configuración. Un `raidtab` de ejemplo se parece a esto:

```
raiddev /dev/md0
        raid-level      0
        nr-raid-disks   2
        persistent-superblock 1
        chunk-size     4
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Como en el modo lineal, los discos de reserva tampoco se soportan aquí. Un RAID-0 no tiene redundancia, por lo que cuando un disco muere, el array le acompaña.

Una vez más, ejecute simplemente

```
mkraid /dev/md0
```

para inicializar el array. Esto debe inicializar los superbloques y poner en funcionamiento el dispositivo RAID. Éche un vistazo a `/proc/mdstat` para ver qué sucede. Debería ver que su dispositivo ahora está en funcionamiento.

`/dev/md0` está listo para ser formateado, montado, usado y maltratado.

4.4 RAID-1

Tiene dos dispositivos de aproximadamente el mismo tamaño y quiere que cada uno de los dos sea un duplicado del otro. Finalmente, tiene más dispositivos que quiere guardar como discos de reserva preparados, que automáticamente formarán parte del duplicado si uno de los dispositivos activos se rompe.

Configure así el fichero `/etc/raidtab`:

```
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        nr-spare-disks  0
        chunk-size      4
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Si tiene discos de reserva, puede añadirlos al final de la especificación de dispositivos como

```
        device          /dev/sdd5
        spare-disk      0
```

Recuerde configurar la entrada `nr-spare-disks` adecuadamente.

De acuerdo, ahora estamos listos para comenzar la inicialización del RAID. Se debe construir el duplicado, es decir, los contenidos (de todos modos, sin importancia ahora, ya que el dispositivo todavía está sin formatear) de los dos dispositivos se deben sincronizar.

Dé la orden

```
mkraid /dev/md0
```

para comenzar la inicialización del duplicado.

Compruebe el fichero `/proc/mdstat`. Debe decirle que se ha puesto en funcionamiento el dispositivo `/dev/md0`, que está siendo reconstruido el duplicado y una cuenta del tiempo estimado para la terminación de la reconstrucción.

La reconstrucción se realiza usando el ancho de banda libre de E/S. De esta manera, su sistema debe ser capaz todavía de responder en gran medida, aunque los LEDs de sus discos deben parpadear lozanamente.

El proceso de reconstrucción es transparente, por lo que realmente puede usar el dispositivo aunque la duplicación esté actualmente en curso.

Intente formatear el dispositivo mientras la reconstrucción se esté realizando. Funcionará. También puede montarlo y usarlo mientras la reconstrucción se esté realizando. Naturalmente, si el disco equivocado se rompe mientras se está realizando la reconstrucción, no hay solución.

4.5 RAID-4

¡Nota! No he comprobado esta configuración por mí mismo. La configuración de más abajo es mi mejor suposición, no algo que realmente haya tenido funcionando.

Tiene tres o más dispositivos de aproximadamente el mismo tamaño, un dispositivo es significativamente más rápido que los otros dispositivos y quiere combinarlos todos en un único dispositivo más grande, conservando todavía alguna información de redundancia. Finalmente, tiene varios dispositivos que desea usar como discos de reserva.

Configure el fichero `/etc/raidtab` así:

```
raiddev /dev/md0
        raid-level      4
        nr-raid-disks   4
        nr-spare-disks  0
        persistent-superblock 1
        chunk-size      32
        device          /dev/sdb1
        raid-disk       0
        device          /dev/sdc1
        raid-disk       1
        device          /dev/sdd1
        raid-disk       2
        device          /dev/sde1
        raid-disk       3
```

Si tuviéramos discos de reserva, se insertarían de forma parecida, siguiendo las especificaciones de discos RAID;

```
        device         /dev/sdf1
        spare-disk     0
```

como de costumbre.

Su array se puede inicializar con la orden

```
   mkraid /dev/md0
```

como es habitual.

Debería ver la sección 4.9 () de opciones especiales de `mke2fs` antes de formatear el dispositivo.

4.6 RAID-5

Tiene tres o más dispositivos de aproximadamente el mismo tamaño, quiere combinarlos en un dispositivo mayor, pero conservando todavía cierto grado de redundancia para la seguridad de datos. Finalmente, tiene varios dispositivos para usar como discos de reserva, que no tomarán parte en el array antes de que otro dispositivo falle.

Si usa N dispositivos donde el tamaño del más pequeño es S, el tamaño de todo el array será *(N-1)*S*. El espacio que falta se usa para información de paridad (redundancia). De esta manera, si cualquier disco falla, todos los datos permanecerán intactos. Pero si dos discos fallan, todos los datos se perderán.

Configure el fichero `/etc/raidtab` así:

```
raiddev /dev/md0
        raid-level      5
        nr-raid-disks   7
        nr-spare-disks  0
        persistent-superblock 1
        parity-algorithm        left-symmetric
        chunk-size      32
        device          /dev/sda3
        raid-disk       0
        device          /dev/sdb1
        raid-disk       1
        device          /dev/sdc1
        raid-disk       2
        device          /dev/sdd1
        raid-disk       3
        device          /dev/sde1
        raid-disk       4
        device          /dev/sdf1
        raid-disk       5
        device          /dev/sdg1
        raid-disk       6
```

Si tuviéramos discos de reserva, se insertarían de forma parecida, siguiendo las especificaciones de discos RAID;

```
        device          /dev/sdh1
        spare-disk      0
```

Y así sucesivamente.

Un tamaño de segmento de 32KB es un buen valor por defecto para muchos sistemas de ficheros de propósito general de estas proporciones. El array sobre el que se utiliza el raidtab anterior es un dispositivo de 7 por 6 GB = 36 GB (recuerde que *(N-1)*S = (7-1)*6 = 36)*. Contiene un sistema de ficheros ext2 con un tamaño de bloque de 4KB. Podría incrementar tanto el tamaño del segmento unitario del array como el tamaño de bloque del sistema de ficheros si su sistema de ficheros fuera o bien mucho mayor o bien si simplemente contuviera ficheros muy grandes.

Vale, ya hemos hablado bastante. Configure el fichero `/etc/raidtab` y veamos si funciona. Ejecute la orden

```
mkraid /dev/md0
```

y observe qué ocurre. Es de esperar que sus discos comiencen a trabajar como locos debido a que empiezan la reconstrucción de su array. Échele un vistazo a `/proc/mdstat` para ver qué está sucediendo.

Si el dispositivo se ha creado correctamente, el proceso de reconstrucción comenzará ahora. Su array no será consistente hasta que esta fase de reconstrucción haya terminado. No obstante, el array es totalmente funcional (excepto, por supuesto, para el manejo de fallos de dispositivos) y puede formatearlo y usarlo incluso mientras se esté reconstruyendo.

Consulte la sección 4.9 () de opciones especiales de `mke2fs` antes de formatear el array.

Bueno, ahora que ya tiene su dispositivo RAID funcionando, siempre puede pararlo o rearrancarlo usando las órdenes

```
raidstop /dev/md0
```

y

```
raidstart /dev/md0,
```

respectivamente.

En lugar de colocar éstos en ficheros de inicio y rearrancar un número astronómico de veces hasta hacer que funcione, siga leyendo y haga funcionar la autodetección.

4.7 El superbloque persistente

Si volviéramos a *aquellos maravillosos días* (*The Good Old Days (TM)*), las herramientas RAID (`raidtools`) leerían su fichero `/etc/raidtab` y a continuación inicializarían el array. Sin embargo, esto requeriría que el sistema de ficheros sobre el que reside `/etc/raidtab` estuviera montado. Esto es imposible si quiere arrancar a partir de un RAID.

También, la anterior aproximación producía complicaciones al montar sistemas de ficheros sobre dispositivos RAID. Éstos no se podían colocar en el fichero `/etc/fstab` como era usual, sino que tenían que ser montados en los guiones (*scripts*) de inicio.

Los superbloques persistentes solucionan estos problemas. Cuando un array se inicializa con la opción `persistent-superblock` en el fichero `/etc/raidtab`, se escribe un superbloque especial al principio de todos los discos participantes en el array. Esto permite al núcleo leer la configuración de los dispositivos RAID directamente de los discos involucrados, en lugar de leerla de algún fichero de configuración que puede no estar disponible en todo momento.

Sin embargo, todavía debería mantener un fichero `/etc/raidtab` consistente, ya que puede necesitar este fichero para una reconstrucción posterior del array.

Los superbloques persistentes son obligatorios si desea la autodetección de sus dispositivos RAID durante el arranque del sistema. Esto se describe en la sección 4.10 ().

4.8 Tamaños de segmento unitario

El tamaño de segmento unitario merece una explicación. Nunca puede escribir de forma totalmente paralela a un grupo de discos. Si tuviera dos discos y quisiera escribir un byte, tendría que escribir cuatro bits en cada disco; realmente, todos los segundos bits irían al disco 0 y los otros al disco 1. Sencillamente, el hardware no soporta eso. En su lugar, elegimos algún tamaño de de segmento unitario que definimos como la masa *atómica* más pequeña de datos que puede ser escrita en los dispositivos. Una escritura de 16 KB con un tamaño de segmento unitario de 4 KB provocaría que el primer y tercer segmento unitario de 4KB se escriban en el primer disco, y el segundo y el cuarto en el segundo, en el caso de un RAID-0 de dos discos. De esta manera, para grandes escrituras, podría observar menor sobrecarga teniendo segmentos lo suficientemente grandes, mientras que los arrays que contuvieran principalmente ficheros pequeños se podrían beneficiar más de un tamaño de segmento unitario más pequeño.

Los tamaños de segmento unitario se pueden especificar para todos los niveles de RAID excepto para el modo lineal.

Para un rendimiento óptimo, debería experimentar con el valor, así como con el tamaño de bloque del sistema de ficheros que pusiera en el array.

El argumento de la opción `chunk-size` en `/etc/raidtab` especifica el tamaño de segmento unitario en kilobytes. Por tanto, 4 significa *4 KB*.

4.8.1 RAID-0

Los datos se escriben *casi* en paralelo en todos los discos del array. Realmente, se escriben `chunk-size` bytes en cada disco, de forma consecutiva.

Si especifica un tamaño de segmento unitario de 4 KB y escribe 16 KB a un array de 3 discos, el sistema RAID escribirá 4 KB a los discos 0, 1 y 2, en paralelo, y a continuación los 4 KB restantes al disco 0.

Un tamaño de segmento unitario de 32 KB es un punto de partida razonable para la mayoría de los arrays. Pero el valor óptimo depende muchísimo del número de discos implicados, del contenido del sistema de ficheros que aloja y de muchos otros factores. Experimente con él para obtener el mejor rendimiento.

4.8.2 RAID-1

Para las escrituras, el tamaño de segmento unitario no afecta al array, ya que se deben escribir todos los datos a todos los discos sin importar qué. Para las lecturas, sin embargo, el tamaño de segmento unitario indica cuántos datos leer consecutivamente de los discos participantes. Ya que todos los discos activos del array contienen la misma información, las lecturas se pueden hacer en paralelo al estilo de un RAID-0.

4.8.3 RAID-4

Cuando se realiza una escritura en un array RAID-4, también se debe actualizar la información de paridad en el disco de paridad. El tamaño de segmento unitario es el tamaño de los bloques de paridad. Si se escribe un byte a un array RAID-4, entonces se leerán `chunk-size` bytes de los *N-1* discos, se calculará la información de paridad y se escribirán `chunk-size` bytes al disco de paridad.

El tamaño de segmento unitario afecta al rendimiento de las lecturas de la misma manera que en un RAID-0, ya que las lecturas de un RAID-4 se realizan de la misma forma.

4.8.4 RAID-5

En RAID-5 el tamaño de segmento unitario tiene exactamente el mismo significado que en un RAID-4.

Un tamaño de segmento unitario razonable para un RAID-5 es 128 KB pero, como siempre, puede desear experimentar con éste.

También consulte la sección 4.9 () de opciones especiales de mke2fs. Esto afecta al rendimiento de un RAID-5.

4.9 Opciones de `mke2fs`

Hay disponible una opción especial cuando se formatean dispositivos RAID-4 y RAID-5 con `mke2fs`. La opción `-R stride=nn` permitirá a `mke2fs` situar mejor diferentes estructuras de datos específicas de ext2 en un dispositivo RAID de forma inteligente.

Si el tamaño de segmento unitario es 32 KB significa que 32 KB de datos consecutivos residirán en un único disco. Si queremos construir un sistema de ficheros ext2 con un tamaño de bloque de 4KB, nos damos cuenta de que habrá 8 bloques del sistema de ficheros en una segmento unitario del array. Podemos pasar esta información a la utilidad mke2fs cuando se cree el sistema de ficheros:

```
mke2fs -b 4096 -R stride=8 /dev/md0
```

El rendimiento de un RAID-{4,5} se ve fuertemente influido por esta opción. No estoy seguro de cómo la opción `stride` afectará a otros niveles RAID. Si alguien tiene información sobre esto, por favor, que la envíe a mi dirección email.

4.10 Autodetección

La autodetección permite a los dispositivos RAID ser automáticamente reconocidos por el núcleo durante el arranque, justo después de que se realice la detección ordinaria de particiones.

Esto requiere varias cosas:

1. Necesita todo el soporte necesario para autodetección (SCSI, IDE, RAID, etc) en el núcleo (no como módulo) o bien crear una imagen `initrd`, lo cual de todos modos es absurdo, dado lo improbable de que descargue dicho soporte en un sistema activo.

   Compruebe esto.

2. Debe haber creado los dispositivos RAID usando superbloques persistentes.

3. El tipo de partición de los dispositivos usados en el RAID se debe establecer a `0xFD` (use `fdisk` y establezca el tipo a `fd`)

NOTA: asegúrese de que su **RAID NO ESTÁ FUNCIONANDO** antes de cambiar los tipos de las particiones. Use `raidstop /dev/md0` para parar el dispositivo.

Si sigue los pasos 1, 2 y 3 de arriba, la autodetección debería activarse. Pruebe rearrancar. Cuando el sistema se levante, vea el contenido de `/proc/mdstat`; debería decirle que su RAID está funcionando.

Durante el arranque, podría ver mensajes similares a éstos:

```
Oct 22 00:51:59 malthe kernel: SCSI device sdg: hdwr sector= 512
 bytes. Sectors= 12657717 [6180 MB] [6.2 GB]
Oct 22 00:51:59 malthe kernel: Partition check:
Oct 22 00:51:59 malthe kernel:  sda: sda1 sda2 sda3 sda4
Oct 22 00:51:59 malthe kernel:  sdb: sdb1 sdb2
Oct 22 00:51:59 malthe kernel:  sdc: sdc1 sdc2
Oct 22 00:51:59 malthe kernel:  sdd: sdd1 sdd2
Oct 22 00:51:59 malthe kernel:  sde: sde1 sde2
Oct 22 00:51:59 malthe kernel:  sdf: sdf1 sdf2
Oct 22 00:51:59 malthe kernel:  sdg: sdg1 sdg2
Oct 22 00:51:59 malthe kernel: autodetecting RAID arrays
Oct 22 00:51:59 malthe kernel: (read) sdb1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdb1,1>
Oct 22 00:51:59 malthe kernel: (read) sdc1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdc1,2>
Oct 22 00:51:59 malthe kernel: (read) sdd1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdd1,3>
Oct 22 00:51:59 malthe kernel: (read) sde1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sde1,4>
Oct 22 00:51:59 malthe kernel: (read) sdf1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdf1,5>
Oct 22 00:51:59 malthe kernel: (read) sdg1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdg1,6>
Oct 22 00:51:59 malthe kernel: autorunning md0
Oct 22 00:51:59 malthe kernel: running: <sdg1><sdf1><sde1><sdd1><sdc1><sdb1>
Oct 22 00:51:59 malthe kernel: now!
```

```
Oct 22 00:51:59 malthe kernel: md: md0: raid array is not clean --
 starting background reconstruction
```

Esta es la salida de la autodetección de un array RAID-5 que no fue limpiamente desactivado (es decir, la máquina se cayó). La reconstrucción se inicia automáticamente. Montar este dispositivo es perfectamente seguro, ya que la reconstrucción es transparente y todos los datos son consistentes (sólo es la información de paridad la que es inconsistente - aunque la misma no se necesita hasta que un dispositivo falle).

Los dispositivos autoarrancados también son automáticamente parados durante el cierre del sistema. No se preocupe de los guiones de inicio. Simplemente, use los dispositivos `/dev/md` como cualquier otro dispositivo `/dev/sdX` o `/dev/hdX`.

Sí, verdaderamente es así de fácil.

Quizás desee buscar cualquier orden `raidstart/raidstop` en sus guiones de inicio/parada `/etc/rc.d/rc.sysinit` y `/etc/rc.d/init.d/halt` (guiones de inicio estándares de RedHat). Se usan para el antiguo estilo de RAID y no tienen utilidad en el nuevo estilo de RAID con autodetección. Elimine las líneas y todo irá perfectamente bien.

4.11 Arrancar desde RAID

Existen varias formas de configurar un sistema que monta su sistema de ficheros raíz sobre un dispositivo RAID. Desafortunadamente, ninguna de las distribuciones de Linux con las que yo he probado (RedHat y Debian) soportan un dispositivo RAID como dispositivo del sistema de ficheros raíz durante el proceso de instalación. Por tanto, le va a doler un poco si quiere esto pero, de hecho, es posible.

Actualmente, LILO no maneja dispositivos RAID y por ello, no se puede cargar el núcleo desde un dispositivo RAID en el instante del arranque. Su sistema de ficheros `/boot` tendrá que residir en un dispositivo que no sea RAID. Un modo de asegurar que su sistema arranca, pase lo que pase, es crear particiones `/boot` similares en todas las unidades de su RAID, de esa forma la BIOS siempre puede cargar datos desde, por ejemplo, la primera unidad disponible. Esto necesita que no arranque con un disco defectuoso en su sistema.

Otra forma de asegurar que su sistema siempre arranca es crear un disquete de arranque cuando toda la configuración se haya terminado. Si muere el disco en el que reside el sistema de ficheros `/boot`, siempre puede arrancar desde el disquete.

4.11.1 Método 1

Este método asume que posee un disco de reserva en el que puede instalar el sistema y que no es parte del RAID que configurará más adelante.

- Primero, instale un sistema normal en su disco extra.
- Obtenga el núcleo que piensa ejecutar, obtenga los parches y las herramientas RAID y haga que su sistema arranque con el nuevo núcleo con soporte RAID. Asegúrese de que el soporte RAID está **dentro** del núcleo y que no se carga como módulo.
- Ahora debe configurar y crear el RAID que tiene pensado usar para el sistema de ficheros raíz. Éste es un procedimiento estándar como ya se describió en 4 ().
- Simplemente para asegurarse de que todo está bien, trate de rearrancar el sistema para ver si el nuevo RAID aparece durante el arranque. Debería aparecer.
- Coloque un sistema de ficheros sobre el nuevo array (usando `mke2fs`), y móntelo en `/mnt/newroot`.

- Ahora, copie el contenido de su sistema de ficheros raíz actual (el disco extra) al nuevo sistema de ficheros raíz (el array). Hay muchas formas de hacer esto. Una de ellas es

```
cd /
find . -xdev | cpio -pm /mnt/newroot
```

- Debe modificar el fichero `/mnt/newroot/etc/fstab` para usar el dispositivo correcto (el dispositivo raíz `/dev/md?`) para el sistema de ficheros raíz.
- Ahora, desmonte el sistema de ficheros `/boot` actual y móntelo en su lugar en `/mnt/newroot/boot`. Esto es necesario para que LILO funcione correctamente en el siguiente paso.
- Actualice `/mnt/newroot/etc/lilo.conf` para que apunte a los dispositivos correctos. El dispositivo de arranque debe ser todavía un disco normal (no un dispositivo RAID) pero el dispositivo raíz debe apuntar a su nuevo RAID. Cuando esté hecho, ejecute `lilo -r /mnt/newroot` Esta ejecución de LILO debería terminar sin errores.
- Rearranque el sistema y observe que todo aparece como se esperaba :)

Si está haciendo esto con discos IDE, asegúrese de indicarle a su BIOS que todos los discos son del tipo «auto-detect», así la BIOS permitirá a su máquina arrancar incluso cuando un disco haya fallado.

4.11.2 Método 2

Este método necesita que parchee su paquete `raidtools` para poder incluir la directiva `failed-disk` en `/etc/raidtab`. Busque en los archivos de la lista de correo Linux-raid los mensajes enviados por Martin Bene, alrededor del 23 de abril de 1999, donde se envió el parche `failed-disk`. Se espera que esta funcionalidad esté pronto en el paquete `raidtools` (para cuando esté leyendo esto puede que incluso no necesite parchear las `raidtools`).

Sólo puede utilizar este método en RAIDs de niveles 1 o superiores. La idea es instalar un sistema sobre un disco que es adrede marcado como estropeado en el RAID, copiar a continuación el sistema en el RAID que estará funcionando en modo degrado y finalmente hacer que el RAID use el ya no necesario `disco de instalación`, aniquilando la anterior instalación pero haciendo que el RAID funcione en modo no degradado.

- Primero, instale un sistema normal sobre un disco (que más tarde formará parte de su RAID). ¡Es importante que este disco (o partición) no sea el más pequeño. Si lo es, no será posible añadirlo al RAID más tarde!
- A continuación, obtenga el núcleo, los parches, las herramientas, etc., etc. Ya conoce el ejercicio. Haga que su sistema arranque con un nuevo núcleo que tenga el soporte RAID que necesita compilado dentro del núcleo.
- Ahora, configure el RAID con su dispositivo raíz actual como el `failed-disk` (disco estropeado) en el fichero `raidtab`. No coloque el `failed-disk` como el primer disco en el fichero `raidtab`, eso le dará problemas para poner en marcha el RAID. Cree el RAID y coloque un sistema de ficheros en él.
- Pruebe a rearrancar y vea si el RAID aparece como debería hacerlo.
- Copie los ficheros del sistema y reconfigure el sistema para usar el RAID como dispositivo raíz, como se ha descrito en la sección 4.11.1 () anterior.
- Cuando su sistema arranque con éxito desde el RAID, puede modificar el fichero `raidtab` para incluir el `failed-disk` anterior como un disco `raid-disk` normal. Ahora, ejecute `raidhotadd` para añadir el disco a su sistema RAID.
- Ahora debería tener un sistema capaz de arrancar desde un RAID no degradado.

4.12 Dificultades

Nunca NUNCA **nunca** reparticione discos que son parte de un RAID que está funcionando. Si debe alterar la tabla de particiones de un disco que es parte de un RAID, pare primero el array y reparticione después.

Es fácil poner demasiados discos en un bus. Un bus Fast-Wide SCSI normal puede sostener 10 MB/s que es menos de lo que muchos discos pueden obtener por sí solos hoy en día. Por supuesto, colocar seis de tales discos en un bus no le proporcionará el aumento de rendimiento esperado.

La mayoría de los controladores SCSI sólo le proporcionarán un rendimiento extra si los buses SCSI son llevados prácticamente al máximo por los discos conectados a ellos. No observará una mejora de rendimiento por usar dos controladoras 2940 con dos discos SCSI viejos en lugar de simplemente hacer funcionar los dos discos sobre una sola tarjeta.

Si olvida la opción `persistent-superblock` puede que su array no arranque por las buenas después de que haya sido parado. Simplemente, recree el array con la opción colocada correctamente en el fichero `/etc/raidtab`.

Si un RAID-5 no logra reconstruirse después de que un disco haya sido eliminado y reinsertado, puede deberse al orden de los dispositivos en el fichero `/etc/raidtab`. Intente mover el primer par `device - raid-disk` al final de la descripción del array en el fichero raidtab.

5 Comprobación

Si piensa usar un RAID para obtener tolerancia a fallos, también puede que quiera comprobar su configuración para ver si realmente funciona. Ahora bien, ¿cómo se simula un fallo de disco?.

El resumen es que no puede, salvo quizás atravesando mediante un hacha incandescente la unidad sobre la que quiere *simular* el fallo. Nunca puede saber qué ocurrirá si un disco muere. Puede que se apodere eléctricamente del bus al que está conectado, haciendo que todas las unidades de ese bus sean inaccesibles, aunque nunca he oído que eso haya ocurrido. La unidad también puede simplemente informar de un fallo de lectura/escritura a la capa SCSI/IDE que a su vez hará que la capa RAID maneje esta situación de forma elegante. Afortunadamente, esta es la forma en la que normalmente ocurren las cosas.

5.1 Simulación de un fallo de disco

Si quiere simular un fallo de disco desconecte la unidad. Debe hacer esto con el **sistema apagado**. Si está interesado en comprobar si sus datos pueden sobrevivir con un disco menos de los habituales, no hay motivo para ser un vaquero de las conexiones en caliente aquí. Apague el sistema, desconecte el disco y enciéndalo de nuevo.

Mire en el registro del sistema (generado por `syslogd`) y en `/proc/mdstat` para ver qué es lo que está haciendo el RAID. ¿Ha funcionado?.

Recuerde que **debe** utilizar un RAID-{1,4,5} para que su array sea capaz de sobrevivir a un fallo de disco. Un modo lineal o un RAID-0 fallarán totalmente cuando se pierda un dispositivo.

Cuando haya reconectado el disco de nuevo (recuerde, con el sistema apagado, naturalmente) podrá añadir el *nuevo* dispositivo al RAID otra vez, con la orden `raidhotadd`.

5.2 Simulación de corrupción de datos

Un RAID (ya sea hardware o software) asume que si una escritura en un disco no devuelve un error, entonces la escritura ha tenido éxito. Por tanto. si su disco corrompe datos sin devolver un error, sus datos se *corromperán*. Naturalmente, esto es muy improbable que ocurra, pero es posible, y produciría un sistema de ficheros corrupto.

Un RAID no puede y no está pensado para proteger contra la corrupción de datos del medio de almacenamiento físico. Por tanto, tampoco tiene ningún sentido corromper a propósito los datos de un disco (usando `dd`, por ejemplo) para ver cómo manejará el sistema RAID esa situación. Es más probable (a menos que corrompa el superbloque del RAID) que la capa RAID no descubra nunca la corrupción, sino que su sistema de ficheros en el dispositivo RAID se corrompa.

Así es como se supone que funcionan las cosas. Un RAID no es una garantía absoluta para la integridad de datos, simplemente le permite conservar sus datos si un disco muere (naturalmente, con RAIDs de niveles iguales o superiores a 1).

6 Rendimiento

Esta sección contiene varias pruebas de evaluación de prestaciones (*benchmarks*) de un sistema real usando un RAID software.

Las evaluaciones se han realizado con el programa `bonnie` y todas las veces con ficheros dos o más veces más grandes que el tamaño de la RAM física de la máquina.

Estas evaluaciones *sólo* miden el ancho de banda de entrada y de salida sobre un único gran fichero. Esto es algo interesante de saber si uno está interesado en el máximo rendimiento de E/S para grandes lecturas/escrituras. Sin embargo, tales números nos dicen poco sobre cuál sería el rendimiento si el array se usara para un almacén temporal de noticias, un servidor web, etc. etc. Tenga siempre en cuenta que los números de las evaluaciones son el resultado de ejecutar un programa *sintético*. Pocos programas del mundo real hacen lo que `bonnie` hace y, aunque es interesante mirar estos números de E/S, no son indicadores en última instancia del rendimiento de los dispositivos del mundo real.

Por ahora, sólo poseo resultados de mi propia máquina. La configuración es:

- Dual Pentium Pro 150 MHz
- 256 MB RAM (60 MHz EDO)
- Tres IBM UltraStar 9ES 4.5 GB U2W SCSI
- Adaptec 2940U2W
- Un IBM UltraStar 9ES 4.5 GB UW SCSI
- Adaptec 2940 UW
- Núcleo 2.2.7 con los parches RAID

Los tres discos U2W cuelgan de la controladora U2W y el disco UW cuelga de la controladora UW.

Parece imposible sacar mucho más de 30 MB/s a través de los buses SCSI de este sistema, usando un RAID o no. Mi suposición es que, debido a que el sistema es bastante antiguo, el ancho de banda de la memoria lo fastidia y, por tanto, limita lo que se puede enviar a través de las controladoras SCSI.

6.1 RAID-0

Lectura significa **entrada de bloques secuencial** y **Escritura** significa **salida de bloques secuencial**. El tamaño de fichero fue de 1GB en todas las pruebas. Las pruebas se realizaron en modo monousuario. Se configuró el controlador (*driver*) SCSI para que no utilizara *tagged command queuing*, *TCQ*).

Tamaño de segmento unitario	Tamaño de bloque	Lectura KB/s	Escritura KB/s
4k	1k	19712	18035
4k	4k	34048	27061
8k	1k	19301	18091
8k	4k	33920	27118
16k	1k	19330	18179
16k	2k	28161	23682
16k	4k	33990	27229
32k	1k	19251	18194
32k	4k	34071	26976

A partir de esto vemos que el tamaño de segmento unitario del RAID no importa mucho. Sin embargo, el tamaño de bloque del sistema de ficheros ext2 debería ser tan grande como fuera posible, lo cual significa 4KB (es decir, el tamaño de página) en una IA-32 (N.T.: arquitectura Intel de 32 bits).

6.2 RAID-0 con TCQ

Esta vez, el manejador SCSI se configuró para usar *TCQ*, con una longitud de cola de 8. Por lo demás, todo es lo mismo de antes.

Tamaño de segmento unitario	Tamaño de bloque	Lectura KB/s	Escritura KB/s
32k	4k	33617	27215

No se realizaron más pruebas. Activar el *TCQ* pareció incrementar ligeramente el rendimiento de las escrituras, pero verdaderamente no hubo mucha diferencia en absoluto.

6.3 RAID-5

El array se configuró para funcionar en el modo RAID-5 y se hicieron pruebas similares.

Tamaño de segmento unitario	Tamaño de bloque	Lectura KB/s	Escritura KB/s
8k	1k	11090	6874
8k	4k	13474	12229
32k	1k	11442	8291
32k	2k	16089	10926
32k	4k	18724	12627

Ahora, tanto el tamaño de segmento unitario como el tamaño de bloque parecen realmente significativos.

6.4 RAID-10

Un RAID-10 significa *bandas duplicadas* o un array RAID-1 de dos arrays RAID-0. El tamaño de segmento unitario es tanto el tamaño de las porciones del array RAID-1 como del array RAID-0. No realicé pruebas en las que esos tamaños de segmento unitario fueran diferentes, aunque esa debería ser una configuración perfectamente válida.

Tamaño de segmento unitario	Tamaño de bloque	Lectura KB/s	Escritura KB/s
32k	1k	13753	11580
32k	4k	23432	22249

No se realizaron más pruebas. El tamaño de fichero fue de 900MB debido a que las cuatro particiones involucradas eran de 500 MB cada una, lo cual no deja espacio para un fichero de 1GB en esta configuración (RAID-1 sobre dos arrays de 1000MB).

7 Agradecimientos

Las siguientes personas han contribuido a la creación de este documento:

- Ingo Molnar
- Jim Warren
- Louis Mandelstam
- Allan Noah
- Yasunori Taniike
- La gente de la lista de correo Linux-RAID
- El que se me olvida, lo siento :)

Por favor, envíe correcciones, sugerencias, etc. al autor. Es la única forma en que este CÓMO puede mejorar.

Envíe correcciones, sugerencias, etc. sobre esta traducción al español a Juan Piernas Cánovas (*piernas@ditec.um.es*).

8 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. :)), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones «oficiales»: *www.insflug.org*. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Además, cuenta con un sistema interactivo de gestión de fe de erratas y sugerencias en línea, motor de búsqueda específico, y más servicios que estamos trabajando incesantemente para añadir.

Se proporcionará también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

en ***http://www.insflug.org/insflug/creditos.php3*** cuenta con una detallada relación de las personas que hacen posible tanto esto como las traducciones.

Diríjase a *http://www.insflug.org/colaboracion/index.php3* si desea unirse a nosotros.

Francisco José Montilla, *pacopepe@insflug.org*.