Guía de administración de Oracle Solaris ZFS

Referencia: 820–2314–15
Septiembre de 2010

ORACLE

100901@24661

Contenido

Prefacio

La *Guía de administración de Oracle Solaris ZFS* proporciona información sobre la configuración y administración de sistemas de archivos ZFS de Oracle Solaris.

Esta guía contiene información para los sistemas basados en SPARC y x86.

Nota – Esta versión de Oracle Solaris es compatible con sistemas que usen arquitecturas de las familias de procesadores SPARC y x86: UltraSPARC, SPARC64, AMD64, Pentium y Xeon EM64T. Los sistemas admitidos aparecen en la *Lista de compatibilidad de hardware de Solaris 10* en http://www.sun.com/bigadmin/hcl. Este documento indica las diferencias de implementación entre los tipos de plataforma.

En el presente documento, estos términos x86 significan lo siguiente:

- "x86" hace referencia a la familia más grande de productos compatibles con 64 y 32 bits.
- "x64" destaca información específica de 64 bits acerca de los sistemas AMD64 o EM64T.
- "x86 de 32 bits" destaca información específica de 32 bits acerca de sistemas basados en x86.

Para conocer cuáles son los sistemas admitidos, consulte la *Lista de compatibilidad de hardware de Solaris 10*.

Quién debe utilizar este manual

Esta guía va dirigida a los usuarios interesados en la configuración y administración de los sistemas de archivos ZFS de Oracle Solaris. Es aconsejable tener experiencia previa con el sistema operativo (SO) Oracle Solaris u otra versión de UNIX.

Organización de este manual

En la tabla siguiente se describen los capítulos de este manual.

Capítulo	Descripción
Capítulo 1, “Sistema de archivos ZFS de Oracle Solaris (introducción)”	Ofrece una descripción general de ZFS, sus características y ventajas. También abarca la terminología y algunos conceptos básicos.
Capítulo 2, “Procedimientos iniciales con Oracle Solaris ZFS ”	Ofrece instrucciones paso a paso para configuraciones ZFS sencillas con sistemas de archivos y agrupaciones simples. Este capítulo también brinda instrucciones de hardware y software necesarias para crear sistemas de archivos ZFS.
Capítulo 3, “Oracle Solaris ZFS y sistemas de archivos tradicionales”	Identifica características importantes que hacen que ZFS sea significativamente diferente respecto a los sistemas de archivos tradicionales. Conocer estas diferencias fundamentales ayudará a solventar dudas al usar herramientas tradicionales junto con ZFS.
Capítulo 4, “Administración de agrupaciones de almacenamiento de Oracle Solaris ZFS”	Proporciona instrucciones detalladas para crear y administrar agrupaciones de almacenamiento de ZFS.
Capítulo 5, “Instalación e inicio de un sistema de archivos raíz ZFS Oracle Solaris”	Describe el procedimiento para instalar e iniciar un sistema de archivos ZFS. También se describe la migración de un sistema de archivos raíz UFS a un sistema de archivos raíz ZFS mediante Actualización automática de Oracle Solaris.
Capítulo 6, “Administrar sistemas de archivos ZFS de Oracle Solaris”	Ofrece información detallada sobre la administración de sistemas de archivos ZFS. Abarca conceptos como la disposición jerárquica del sistema de archivos, la herencia de propiedades, la administración de puntos de montaje automático y cómo compartir interacciones.
Capítulo 7, “Uso de clones e instantáneas de Oracle Solaris ZFS”	Describe cómo crear y administrar clones e instantáneas de ZFS.
Capítulo 8, “Uso de listas ACL para proteger archivos ZFS”	Describe cómo utilizar las listas de control de acceso (ACL) para proteger los archivos ZFS ofreciendo más permisos granulares que los UNIX estándar.
Capítulo 9, “Administración delegada de ZFS”	Describe la forma de utilizar la administración delegada de ZFS para permitir que los usuarios sin privilegios puedan efectuar tareas de administración de ZFS.
Capítulo 10, “Temas avanzados de Oracle Solaris ZFS”	Ofrece información sobre el uso de volúmenes de ZFS, el uso de ZFS en un sistema Oracle Solaris con zonas instaladas y agrupaciones raíz alternativas.

Capítulo	Descripción
Capítulo 11, "Recuperación de agrupaciones y solución de problemas de Oracle Solaris ZFS"	Describe cómo identificar errores de ZFS y cómo resolverlos. También se proporciona pasos para evitar errores.
Apéndice A, "Descripciones de versiones de Oracle Solaris ZFS"	Describe versiones de ZFS disponibles, las características de cada versión y el sistema operativo Solaris pertinente.

Manuales relacionados

Se puede encontrar información relacionada con temas generales de administración del sistema Oracle Solaris en los manuales siguientes:

- *System Administration Guide: Basic Administration*
- *System Administration Guide: Advanced Administration*
- *System Administration Guide: Devices and File Systems*
- *System Administration Guide: Security Services*

Documentación, asistencia y formación

Encontrará recursos adicionales en estos sitios web:

- Documentación (`http://docs.sun.com`)
- Asistencia (`http://www.oracle.com/us/support/systems/index.html`)
- Formación (`http://education.oracle.com`): haga clic en el vínculo de Sun situado en la barra de navegación izquierda.

Oracle valora sus comentarios

Oracle valora sus comentarios y sugerencias sobre la calidad y la utilidad de su documentación. Si encuentra algún error o quiere sugerir alguna mejora, acceda a `http://docs.sun.com` y haga clic en Feedback. Indique el título y el número de referencia de la documentación, y también el capítulo, la sección y el número de la página pertinentes, si es posible. Indique si desea una respuesta.

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) ofrece diversos recursos relacionados con el software Oracle:

- Para discutir problemas técnicos y sus soluciones, utilice los foros de discusión (`http://forums.oracle.com`).
- Para practicar procedimientos paso a paso, utilice Oracle By Example (`http://www.oracle.com/technology/obe/start/index.html`).
- Puede descargar código de muestra (`http://www.oracle.com/technology/sample_code/index.html`).

Convenciones tipográficas

La siguiente tabla describe las convenciones tipográficas utilizadas en este manual.

TABLA P–1 Convenciones tipográficas

Tipos de letra	Significado	Ejemplo
`AaBbCc123`	Los nombres de los comandos, los archivos, los directorios y los resultados que el equipo muestra en pantalla.	Edite el archivo `.login`. Utilice el comando `ls -a` para mostrar todos los archivos. `nombre_sistema% tiene correo.`
`AaBbCc123`	Lo que se escribe, en contraposición con la salida del equipo en pantalla	`nombre_sistema%` **`su`** `Contraseña:`
aabbcc123	Marcador de posición: sustituir por un valor o nombre real	El comando necesario para eliminar un archivo es `rm` *nombrearchivo*.
AaBbCc123	Títulos de los manuales, términos nuevos y palabras destacables	Consulte el capítulo 6 de la *Guía del usuario*. Una *copia en caché* es aquella que se almacena localmente. *No* guarde el archivo. **Nota:** algunos elementos destacados aparecen en negrita en línea.

Indicadores de los shells en los ejemplos de comandos

La tabla siguiente muestra los indicadores de sistema UNIX predeterminados y el indicador de superusuario de shells que se incluyen en los sistemas operativos Oracle Solaris. Tenga en cuenta que el indicador predeterminado del sistema que se muestra en los ejemplos de comandos varía según la versión de Oracle Solaris.

TABLA P–2 Indicadores de shell

Shell	Indicador
Shell Bash, shell Korn y shell Bourne	`$`
Bash, Korn y Bourne son intérpretes de comandos de superusuario	`#`
Shell C	`nombre_sistema%`
Shell C para superusuario	`nombre_sistema#`

◆ ◆ ◆ CAPÍTULO 1

Sistema de archivos ZFS de Oracle Solaris (introducción)

Este capítulo ofrece una visión general del sistema de archivos ZFS de Oracle Solaris, así como de sus funciones y ventajas. También aborda terminología básica utilizada en el resto del manual.

Este capítulo se divide en las secciones siguientes:

Novedades de ZFS

Esta sección resume las funciones nuevas del sistema de archivos ZFS.

División de una agrupación de almacenamiento de ZFS refleja (`zpool split`)

Versión Oracle Solaris 10 9/10: en esta versión de Solaris, puede utilizar el comando `zpool split` para dividir una agrupación de almacenamiento reflejada, que desconecta discos en la agrupación reflejada original para crear otra agrupación idéntica.

Para obtener más información, consulte "Creación de una nueva agrupación mediante la división de una agrupación de almacenamiento de ZFS reflejada" en la página 90.

Nuevo proceso del sistema ZFS

Versión Oracle Solaris 10 9/10: en esta versión de Solaris, cada agrupación de almacenamiento de ZFS tiene un proceso asociado, `zpool-`*poolname* . Los subprocesos de este proceso son los del procesamiento de E/S de la agrupación para manejar tareas E/S, como, la compresión y sumas de comprobación, que están asociadas con la agrupación. La finalidad de este proceso es proporcionar la visibilidad en cada agrupación de almacenamiento del uso de la CPU. La información acerca de estos procesos puede ser revisada utilizando los comandos `ps` y `prstat`. Estos procesos sólo están disponibles en la zona global. Para obtener más información, consulte `SDC(7)`.

Cambios en el comando `zpool list`

Versión Oracle Solaris 10 9/10: en esta versión de Solaris, la salida `zpool list` se ha modificado para ofrecer información de mayor calidad sobre la asignación de espacio. Por ejemplo:

```
# zpool list tank
NAME    SIZE  ALLOC   FREE    CAP  HEALTH  ALTROOT
tank    136G  55.2G  80.8G    40%  ONLINE  -
```

Los campos `USED` y `AVAIL` anterior se han sustituido por `ALLOC` y `FREE`.

El campo `ALLOC` identifica la cantidad de espacio físico asignado a todos los conjuntos de datos y los metadatos internos. El campo `FREE` identifica la cantidad de espacio sin asignar en la agrupación.

Para obtener más información, consulte "Visualización de información de agrupaciones de almacenamiento de ZFS" en la página 107.

Recuperación de agrupación de almacenamiento de ZFS

Versión Oracle Solaris 10 9/10: una agrupación de almacenamiento puede sufrir daño si los dispositivos subyacentes no quede disponible, si se produce un fallo del suministro eléctrico, o si un número superior al de dispositivos admitidos no funcionan en una configuración redundante ZFS. Esta versión incluye nuevas funciones de comando para recuperar la agrupación de almacenamiento dañada. Sin embargo, el uso de esta función de recuperación significa que las últimas transacciones que se produjeron antes de la agrupación suministro.

Tanto el `zpool borrar` y `zpool importar` mandatos dan soporte la `-F` opción para posiblemente recuperarse una agrupación dañada. Además, al ejecutar el comando `zpool status`, `zpool clear` o `zpool import`, se informa automáticamente de una agrupación dañada y estos comandos describen cómo recuperar la agrupación.

Para obtener más información, consulte "Reparación de daños en las agrupaciones de almacenamiento de ZFS" en la página 317.

Mejoras en dispositivos de registro ZFS

Versión Oracle Solaris 10 9/10: las siguientes mejoras en los dispositivos de registro están disponibles:

- La propiedad `logbias`: esta propiedad se puede utilizar para proporcionar una pista a ZFS para el manejo síncrono solicitudes para un conjunto de datos específico. Si `logbias` se establece en `latency`, ZFS utiliza los dispositivos de registro independientes de la agrupación, si los hay, para manejar la solicitudes con latencia baja. Si `logbias` se establece en `throughput`, ZFS no utiliza los dispositivos de registro independientes de la agrupación. En su lugar, ZFS optimiza las operaciones síncronas para el rendimiento global de la agrupación y el uso eficiente de recursos. El valor predeterminado es `latency`. En la mayoría de las configuraciones, se recomienda el valor predeterminado. El uso del valor `logbias=throughput` puede mejorar el rendimiento para escribir los archivos de base.
- Eliminación de dispositivo de registro: ahora puede eliminar un dispositivo de registro de una agrupación de almacenamiento de ZFS mediante el comando `zpool remove`. Se puede eliminar un único dispositivo de registro especificando el nombre del dispositivo. Un dispositivo de registro reflejado se puede eliminar mediante la especificación de la duplicación de nivel superior para el registro. Cuando un dispositivo de registro individual se elimina del sistema, se escriben registros de transacciones de ZIL en la agrupación principal.

  Los dispositivos virtuales de nivel superior redundantes se identifican ahora mediante un identificador numérico. Por ejemplo, en una agrupación de almacenamiento reflejada de dos discos, el dispositivo virtual de nivel superior es `mirror-0`.

  Para obtener más información, consulte el Ejemplo 4–3.

RAIDZ de paridad triple (`raidz3`)

Versión Oracle Solaris 10 9/10: en esta versión de Solaris, una configuración de RAID-Z redundante ahora puede tener una paridad sencilla, doble o triple, lo que significa que se pueden asumir uno, dos o tres errores de dispositivos sin que se produzca una pérdida de datos. Puede especificar la palabra clave raidz3 para una configuración de RAID-Z de paridad triple. Para obtener más información, consulte "Creación de una agrupación de almacenamiento de RAID-Z" en la página 74.

Conservación de instantáneas de ZFS

Versión Oracle Solaris 10 9/10: si se implementan diferentes directivas de instantáneas automáticas de manera que las instantáneas más antiguas se destruyen accidentalmente por `zfs receive` porque ya no existen de la parte remitente, debería considerar el uso de la función de retención de instantáneas en esta versión de Solaris.

Mantener pulsada una instantánea impide que se destruya. Además, esta función permite que una instantánea con clones se elimine en espera de la eliminación del último clon mediante el comando `zfs destroy -d`.

Puede retener una instantánea o un conjunto de instantáneas. Por ejemplo, la siguiente sintaxis coloca una etiqueta de retención, `keep`, en `tank/home/cindys/snap@1` .

```
# zfs hold keep tank/home/cindys@snap1
```

Para obtener más información, consulte "Conservación de instantáneas de ZFS " en la página 227.

Mejoras en sustitución de dispositivos ZFS

Versión Oracle Solaris 10 9/10: en esta versión de Solaris, se suministra un evento de sistema o *sysevent* cuando se amplía un dispositivo subyacente. ZFS se ha mejorado para que reconozca dichos eventos y ajusta la agrupación basado en el nuevo tamaño del LUN expandido, según la configuración de la propiedad `autoexpand`. Puede utilizar la propiedad de agrupación `autoexpand` para activar o desactivar la aplicación automática de agrupaciones cuando se recibe un evento de expansión de LUN dinámico.

Estas funciones permiten expandir un LUN y la agrupación resultante puede acceder al espacio ampliado sin tener que exportar e importar la agrupación o reiniciar el sistema.

Por ejemplo, la expansión automática de LUN está habilitada en la agrupación `tank` .

```
# zpool set autoexpand=on tank
```

O, si lo desea, puede crear la agrupación con la propiedad `autoexpand` habilitada.

```
# zpool create -o autoexpand=on tank c1t13d0
```

La propiedad `autoexpand` está desactivada de manera predeterminada, por lo que se puede decidir si se expande o no el LUN.

Un LUN también puede expandir mediante el comando `zpool online -e`. Por ejemplo:

```
# zpool online -e tank c1t6d0
```

O, si lo desea, puede restablecer la propiedad `autoexpand` después que el LUN se conecte o esté disponible mediante la función `zpool reemplazar`. Por ejemplo, la agrupación siguiente se crea con un disco de 8 GB (`c0t0d0`). El disco 8 GB se sustituye por uno de 16 GB (`c1t13d0`), pero el tamaño de la agrupación no se expande hasta que se habilite la propiedad `autoexpand`.

```
# zpool create pool c0t0d0
# zpool list
NAME   SIZE   ALLOC  FREE    CAP   HEALTH  ALTROOT
pool   8.44G  76.5K  8.44G     0%  ONLINE  -
# zpool replace pool c0t0d0 c1t13d0
# zpool list
NAME   SIZE   ALLOC  FREE    CAP   HEALTH  ALTROOT
pool   8.44G  91.5K  8.44G     0%  ONLINE  -
# zpool set autoexpand=on pool
# zpool list
NAME   SIZE   ALLOC  FREE    CAP   HEALTH  ALTROOT
pool   16.8G   91.5K  16.8G    0%  ONLINE  -
```

Otro modo de expandir el LUN en el ejemplo anterior sin la activación de la propiedad `autoexpand` propiedad es utilizando el comando `zpool online -e`, aunque el dispositivo ya esté en línea. Por ejemplo:

```
# zpool create tank c0t0d0
# zpool list tank
NAME   SIZE   ALLOC  FREE    CAP   HEALTH  ALTROOT
tank   8.44G  76.5K  8.44G     0%  ONLINE  -
# zpool replace tank c0t0d0 c1t13d0
# zpool list tank
NAME   SIZE   ALLOC  FREE    CAP   HEALTH  ALTROOT
tank   8.44G  91.5K  8.44G     0%  ONLINE  -
# zpool online -e tank c1t13d0
# zpool list tank
NAME   SIZE   ALLOC  FREE    CAP   HEALTH  ALTROOT
tank   16.8G    90K  16.8G     0%  ONLINE  -
```

Entre las mejoras en la sustitución de dispositivos adicionales en esta versión se encuentran las funciones siguientes:

- En las versiones anteriores, ZFS no podía reemplazar un disco existente con otro disco, o conectar un disco si el disco de repuesto tenía un tamaño ligeramente diferente. En esta versión, reemplazar un disco existente con otro disco, o conectar un nuevo disco nominalmente del mismo tamaño, siempre que la agrupación no esté llena.

- En esta versión, ya no es necesario reiniciar el sistema, o exportar e importar una agrupación, para expandir un LUN. Como se describe anteriormente, puede habilitar la propiedad `autoexpand` o utilizar el comando `zpool online -e` para expandir el tamaño total de un LUN.

Para obtener más información sobre la sustitución de dispositivos, consulte "Sustitución de dispositivos en una agrupación de almacenamiento" en la página 96.

Compatibilidad con la instalación de ZFS y Flash

Versión Solaris 10 10/09: en esta versión de Solaris puede establecer un perfil JumpStart para identificar un archivo de almacenamiento Flash de una agrupación raíz ZFS. Para obtener más información, consulte "Instalación de un sistema de archivos raíz ZFS (instalación de archivo de almacenamiento flash de Oracle Solaris)" en la página 137.

Cuotas de grupo y usuario de ZFS

Versión Solaris 10 10/09: en las versiones anteriores de Solaris, se podía aplicar cuotas y reservas a los sistemas de archivos ZFS para administrar y reservar espacio en el disco.

En esta versión de Solaris, puede configurar una cuota para la cantidad de espacio en el disco consumido por archivos cuyo propietario es un grupo o usuario determinado. Deberá considerar el establecimiento de cuotas de usuarios o grupos en un entorno con un gran número de usuarios o grupos.

Se puede configurar una cuota de usuarios mediante la propiedad `zfs userquota`. Para configurar una cuota de grupo, utilice la propiedad `zfs groupquota`. Por ejemplo:

```
# zfs set userquota@user1=5G tank/data
# zfs set groupquota@staff=10G tank/staff/admins
```

Puede mostrar la configuración de la cuota actual de un grupo o usuario como se indica a continuación:

```
# zfs get userquota@user1 tank/data
NAME       PROPERTY         VALUE            SOURCE
tank/data  userquota@user1  5G               local
# zfs get groupquota@staff tank/staff/admins
NAME               PROPERTY          VALUE              SOURCE
tank/staff/admins  groupquota@staff  10G                local
```

Visualice información general sobre la cuota como sigue:

```
# zfs userspace tank/data
TYPE        NAME   USED  QUOTA
POSIX User  root     3K   none
POSIX User  user1    0    5G
```

```
# zfs groupspace tank/staff/admins
TYPE         NAME   USED  QUOTA
POSIX Group  root     3K   none
POSIX Group  staff     0    10G
```

Puede mostrar el uso de espacio en el disco de un usuario individual visualizando la propiedad `userused@`*user*. El uso de espacio en el disco de un grupo puede visualizarse con la propiedad `groupused@` *group*. Por ejemplo:

```
# zfs get userused@user1 tank/staff
NAME        PROPERTY        VALUE           SOURCE
tank/staff  userused@user1  213M            local
# zfs get groupused@staff tank/staff
NAME        PROPERTY         VALUE            SOURCE
tank/staff  groupused@staff  213M             local
```

Para obtener más información sobre la configuración de cuotas de usuarios, consulte "Configuración de cuotas y reservas de ZFS" en la página 217.

Herencia de passthrough de listas de control de acceso (ACL) de ZFS para el permiso de ejecución

Versión Solaris 10 10/09: en las versiones anteriores de Solaris podía aplicar herencia de lista de control de acceso (ACL), de forma que todos los archivos se creaban con permisos `0664` o `0666`. En esta versión, si desea incluir de forma opcional el bit de ejecución desde el modo de creación de archivo en la lista de control de acceso (ACL) heredada, puede establecer el modo `aclinherit` para transferir el permiso de ejecución a la lista de control de acceso (ACL) heredada.

Si se ha habilitado `aclinherit=passthrough-x` en un conjunto de datos ZFS, puede incluir el permiso de ejecución para un archivo de salida que se genere desde las herramientas de compilación `cc` o `gcc`. Si la lista de control de acceso (ACL) heredada no incluye permisos de ejecución, la salida ejecutable del compilador no será ejecutable hasta que utilice el comando `chmod` para cambiar los permisos del archivo.

Para obtener más información, consulte el Ejemplo 8–12.

Mejoras en las propiedades de ZFS

Versión Solaris 10/09: las siguientes mejoras en el sistema de archivos ZFS se incluyen en estas versiones.

- **Mejoras en propiedades de flujos instantáneas de ZFS** : puede definir una propiedad recibida que sea diferente de su configuración de propiedad local. Por ejemplo, es posible que reciba un flujo con la propiedad de compresión deshabilita, pero desee habilitar la compresión en el sistema de archivos receptor. Esto significa que el flujo recibido ha

recibido un valor de compresión de `off` y un valor de compresión local de `on`. Dado que el valor local tiene preferencia sobre el valor recibido, no necesita preocuparse por que la configuración de la parte remitente sustituya el valor de la parte receptora. El comando `zfs get` comando muestra el valor efectivo de la propiedad de compresión en la columna `VALUE`.

Las nuevas opciones y propiedades de comandos de ZFS para admitir valores en las propiedades de envío y locales son:

- Utilice `zfs inherit -S` para restablecer un valor de propiedad local al valor recibido, si lo hubiera. Si una propiedad no tiene un valor recibido, el comportamiento del comando `zfs inherit - S` es el mismo que el comando `zfs inherit` sin la opción `-S`. Si la propiedad no tiene un valor recibido, el comando `zfs inherit` enmascara el valor recibido con el valor heredado hasta que la emisión de un comando `zfs inherit -S` lo restablece al valor recibido.
- Puede utilizar `zfs get -o` para incluir la nueva columna `RECEIVED` no predeterminada. O bien, utilice el comando `zfs get -o all` para incluir todas las columnas, incluida `RECEIVED`.
- Puede utilizar la opción `zfs send -p` para incluir las propiedades en el flujo de envío sin la opción `-R`.

Además, puede utilizar la opción `zfs send -e` para utilizar el último elemento del nombre de instantánea enviado para determinar el nuevo nombre de instantánea. El ejemplo siguiente envía la instantánea `poola/bee/cee@1` al sistema `poold/eee` y sólo utiliza el último elemento (`cee@1`) del nombre de la instantánea para crear el sistema y la instantánea del archivo recibido.

```
# zfs list -rt all poola
NAME              USED  AVAIL  REFER  MOUNTPOINT
poola             134K   134G    23K  /poola
poola/bee          44K   134G    23K  /poola/bee
poola/bee/cee      21K   134G    21K  /poola/bee/cee
poola/bee/cee@1      0      -    21K  -
# zfs send -R poola/bee/cee@1 | zfs receive -e poold/eee
# zfs list -rt all poold
NAME              USED  AVAIL  REFER  MOUNTPOINT
poold             134K   134G    23K  /poold
poold/eee          44K   134G    23K  /poold/eee
poold/eee/cee      21K   134G    21K  /poold/eee/cee
poold/eee/cee@1      0      -    21K  -
```

- **Configuración de las propiedades del sistema de archivos ZFS en el momento de crear la agrupación**: puede definir propiedades del sistema de archivos ZFS cuando se crea una agrupación de almacenamiento. En el ejemplo siguiente, la compresión está habilitada en el sistema de archivos ZFS que se crea cuando se crea la agrupación:

  ```
  # zpool create -O compression=on pool mirror c0t1d0 c0t2d0
  ```

- **Configuración de propiedades de la memoria caché en un sistema de archivos ZFS**: dos nuevas propiedades del sistema de archivos ZFS permiten controlar qué se almacena en la memoria caché en la caché primaria (ARC) o en la caché secundaria (L2ARC). Las propiedades de la caché se establecen como se indica a continuación:

- `primarycache`: controla qué se almacena en la memoria caché en la ARC.
- `secondarycache`: controla qué se almacena en la memoria caché en la L2ARC.
- Los valores posibles para ambas propiedades: `all`, `none` y `metadata`. Si se establece en `all`, los datos de usuario y los metadatos se almacenan en la memoria caché. Si se establece en `none`, no se completan datos de usuario ni los metadatos se almacenan en la memoria caché. Si se establece en `metadata`, sólo los metadatos se almacenan en la memoria caché. El valor predeterminado es `all`.

Puede definir estas propiedades en un sistema de archivos existente o cuando se crea el sistema de archivos. Por ejemplo:

```
# zfs set primarycache=metadata tank/datab
# zfs create -o primarycache=metadata tank/newdatab
```

Cuando estas propiedades se establecen en sistemas de archivos existentes, sólo la nueva E/S se basa en la memoria caché en función del valor de estas propiedades.

Algunos entornos de la base de datos pueden beneficiarse de no almacenar datos de usuario en la memoria caché. Se deberá determinar si establecer propiedades de caché es adecuado para su entorno.

- **Visualizar propiedades de cálculo del espacio en el disco**: las nuevas propiedades del sistema de archivos de sólo lectura ayudan a identificar el uso de espacio en el disco para clones, sistemas de archivos, volúmenes e instantáneas. Las propiedades son las siguientes:
  - `usedbychildren`: identifica la cantidad de espacio en el disco utilizado por subordinados de este conjunto de datos, que se liberaría si todos los subordinados del conjunto de datos se destruyeran. La abreviatura de la propiedad es `usedchild`.
  - `usedbydataset`: identifica la cantidad de espacio en el disco que utiliza este conjunto de datos en sí, que se liberaría si se destruyera el conjunto de datos, después de eliminar primero las instantáneas y los `refreservation`. La abreviatura de la propiedad es `usedds`.
  - `usedbyrefreservation`: identifica la cantidad de espacio en el disco que utiliza un `refreservation` definido en este conjunto de datos, que se liberaría si se eliminara el `refreservation`. La abreviatura de la propiedad es `usedrefreserv`.
  - `Usedbysnapshots`: identifica la cantidad de espacio en el disco consumido por las instantáneas de este conjunto de datos, que se liberaría si todas las instantáneas de este conjunto de datos fueran destruidas. Tenga en cuenta que esto no es simplemente la suma de las propiedades `used` de las instantáneas, ya que varias instantáneas pueden compartir el espacio en el disco. La abreviatura de la propiedad es `usedsnap`.

Estas nuevas propiedades desglosan el valor de la propiedad `used` en los diversos elementos que consumen espacio en el disco. En concreto, el valor de la propiedad `used` se desglosa como sigue:

```
used property = usedbychildren + usedbydataset + usedbyrefreservation + usedbysnapshots
```

Puede ver estas propiedades mediante el comando `zfs list -o space`. Por ejemplo:

```
$ zfs list -o space
NAME                AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
rpool              25.4G  7.79G         0     64K              0      7.79G
rpool/ROOT         25.4G  6.29G         0     18K              0      6.29G
rpool/ROOT/snv_98  25.4G  6.29G         0   6.29G              0          0
rpool/dump         25.4G  1.00G         0   1.00G              0          0
rpool/export       25.4G    38K         0     20K              0        18K
rpool/export/home  25.4G    18K         0     18K              0          0
rpool/swap         25.8G   512M         0    111M           401M          0
```

El comando anterior es equivalente al comando `zfs list - o name,avail,used,usedsnap,usedds,usedrefreserv,usedchild -t filesystem,volume`.

- **Listado de instantáneas**: la propiedad de agrupación `listsnapshots` controla si se muestra la información de la instantánea mediante el comando `list zfs`. El valor predeterminado es `on` , lo que significa que la información de la instantánea se muestra de forma predeterminada.

  Si el sistema dispone de varias instantáneas de ZFS y desea desactivar la visualización de información de instantánea en el comando `zfs list`, desactive la propiedad `listsnapshots` de la siguiente forma:

  ```
  # zpool get listsnapshots pool
  NAME  PROPERTY       VALUE      SOURCE
  pool  listsnapshots  on         default
  # zpool set listsnaps=off pool
  ```

  Si inhabilita la propiedad `listsnapshots` , puede utilizar el comando `zfs list -t snapshots` para mostrar la información de la instantánea. Por ejemplo:

  ```
  # zfs list -t snapshot
  NAME                     USED  AVAIL  REFER  MOUNTPOINT
  pool/home@today           16K      -    22K  -
  pool/home/user1@today      0      -    18K  -
  pool/home/user2@today      0      -    18K  -
  pool/home/user3@today      0      -    18K  -
  ```

Recuperación del dispositivo de registros de ZFS

Versión Solaris 10 10/09: en esta versión, ZFS identifica los errores de intento de registro en la salida del comando `zpool status`. Diagnosis de arquitectura de administración fallida (FMA) informa de dichos errores también. Ambos, ZFS y FMA, describen cómo recuperarse de un error de intento de registro.

Por ejemplo, si el sistema se cierra bruscamente antes de que las operaciones de escritura sincrónica se confirmen en una agrupación con un dispositivo de registro independiente, se muestran mensajes parecidos al siguiente:

```
# zpool status -x
  pool: pool
 state: FAULTED
status: One or more of the intent logs could not be read.
```

```
        Waiting for adminstrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
        or ignore the intent log records by running 'zpool clear'.
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        pool        FAULTED      0     0     0 bad intent log
          mirror    ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
        logs        FAULTED      0     0     0 bad intent log
          c0t5d0    UNAVAIL      0     0     0 cannot open
```

Puede resolver el error del dispositivo de registro como se indica a continuación:

- Sustituya o recupere el dispositivo de registro. En este ejemplo, el dispositivo de registro es `c0t5d0`.
- Vuelva a conectar el dispositivo de registro.

  ```
  # zpool online pool c0t5d0
  ```
- Restablezca la condición de error del dispositivo de registro que presenta errores.

  ```
  # zpool clear pool
  ```

Si desea recuperarse de este error sin reemplazar el dispositivo de registro que presenta errores, puede borrar el error con el comando `zpool clear`. En esta situación, la agrupación no funcionará correctamente y los registros se escribirán en la agrupación principal hasta que se sustituya el dispositivo de registro independiente.

Considere el uso de dispositivos de registro reflejados para evitar los casos de error en el dispositivo de registro.

Uso de dispositivos caché en la agrupación de almacenamiento ZFS

Versión Solaris 10 10/09: en esta versión, cuando crea una agrupación, puede especificar *dispositivos caché* que se utilizan para almacenar en la memoria caché datos de la agrupación de almacenamiento.

Los dispositivos de caché ofrecen un nivel adicional de grabación de datos en caché entre la memoria principal y el disco. El uso de dispositivos caché optimiza el rendimiento con cargas de trabajo de lectura aleatorias de contenido principalmente estático.

Se pueden especificar uno o más dispositivos de caché al crear la agrupación. Por ejemplo:

```
# zpool create pool mirror c0t2d0 c0t4d0 cache c0t0d0
# zpool status pool
  pool: pool
```

```
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        pool        ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            c0t2d0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
        cache
          c0t0d0    ONLINE       0     0     0

errors: No known data errors
```

Tras agregar los dispositivos de la caché, gradualmente se llenan con contenido de la memoria principal. Según el tamaño del dispositivo de la caché, puede llevar más de una hora en llenarse. La capacidad y las lecturas se pueden supervisar con el comando `zpool iostat` del modo siguiente:

```
# zpool iostat -v pool 5
```

Los dispositivos caché se pueden agregar o quitar de una agrupación después de crearse dicha agrupación.

Para obtener más información, consulte "Creación de una agrupación de almacenamiento de ZFS con dispositivos caché" en la página 77 y Ejemplo 4–4.

Migración de zona en un entorno ZFS

Versión Solaris 10 5/09: esta versión amplía la compatibilidad para migrar zonas en un entorno ZFS con Actualización automática de Oracle Solaris. Para obtener más información, consulte "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156.

Si desea obtener una lista de problemas conocidos relacionados con esta versión, consulte las notas sobre la versión de Solaris 10 5/09.

Instalación e inicio de ZFS

Versión Solaris 10 10/08: esta versión permite instalar e iniciar un sistema de archivos raíz ZFS. Para instalar un sistema de archivos raíz ZFS puede optar por la instalación inicial o por la función JumpStart. O puede usar Actualización automática de Oracle Solaris para migrar de un sistema de archivos raíz UFS a uno ZFS. Asimismo, se proporciona compatibilidad de ZFS para dispositivos de intercambio y volcado. Si desea más información, consulte el Capítulo 5, "Instalación e inicio de un sistema de archivos raíz ZFS Oracle Solaris".

Para ver una lista de los problemas conocidos de esta versión, visite el siguiente sitio:

http://hub.opensolaris.org/bin/view/Community+Group+zfs/boot

Consulte también las notas de la versión de Solaris 10 10/08.

Inversión (rollback) de un conjunto de datos sin desmontar

Versión Solaris 10 10/08: esta versión permite deshacer un conjunto de datos sin desmontar por primera vez. Esta función hace que ya no se necesite la opción `zfs rollback -f` para forzar una operación de desmontaje. Ya no se admite la opción `-f`, y, si se especifica, se hace caso omiso de ella.

Mejoras en el comando `zfs send`

Versión Solaris 10 10/08: esta versión aporta las mejoras siguientes en el comando `zfs send`. Mediante este comando, ahora se puede realizar las siguientes tareas:

- Envíe todos los flujos incrementales de una instantánea a una instantánea acumulativa. Por ejemplo:

```
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
pool                      428K  16.5G    20K  /pool
pool/fs                    71K  16.5G    21K  /pool/fs
pool/fs@snapA              16K      -  18.5K  -
pool/fs@snapB              17K      -    20K  -
pool/fs@snapC              17K      -  20.5K  -
pool/fs@snapD                0      -    21K  -
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@combo
```

  Esta sintaxis envíe todas las instantáneas incrementales entre `fs@snapA` a `fs@snapD` a `fs@combo`.

- Envíe un flujo de datos incrementales de la instantánea original para crear un clon. Para que se acepte el flujo incremental, la instantánea original ya debe estar en la parte receptora. Por ejemplo:

```
# zfs send -I pool/fs@snap1 pool/clone@snapA > /snaps/fsclonesnap-I
.
.
# zfs receive -F pool/clone < /snaps/fsclonesnap-I
```

- Envíe un flujo de replicación de todos los sistemas de archivos descendentes, hasta las instantáneas nombradas. Cuando se reciben, se conservan todas las propiedades, las instantáneas, los sistemas de archivos descendientes y los clones. Por ejemplo:

```
# zfs send -R pool/fs@snap > snaps/fs-R
```

  Se ilustra de forma detallada en el Ejemplo 7–1.

- Envíe un flujo de repetición incremental. Por ejemplo:

  ```
  # zfs send -R -[iI] @snapA pool/fs@snapD
  ```

  Se ilustra de forma detallada en el Ejemplo 7–1.

Para obtener más información, consulte "Envío y recepción de flujos de instantáneas ZFS complejos" en la página 237.

Cuotas y reservas de ZFS sólo para datos del sistema de archivos

Versión Solaris 10 10/08: además de las funciones de reserva y cuota de ZFS existentes, al calcular el consumo de espacio en el disco esta versión tiene en cuenta reservas y cuotas de conjuntos de datos que no incluyen descendientes, como instantáneas y clones.

- La propiedad `refquota` fuerza un límite físico en la cantidad de espacio en el disco que un conjunto de datos puede consumir. Este límite físico no incluye el espacio en el disco usado por los descendientes, como instantáneas y clones.
- La propiedad `refreservation` establece la cantidad mínima de espacio en el disco que se garantiza a un conjunto de datos, sin incluir sus descendientes.

Por ejemplo, puede establecer un límite de `refquota` de 10 GB para `studentA` que establezca un límite físico de 10 GB de espacio en el disco *referenciado*. Si desea una flexibilidad adicional, puede establecer una cuota de 20 GB que permita administrar instantáneas de `studentA`.

```
# zfs set refquota=10g tank/studentA
# zfs set quota=20g tank/studentA
```

Para obtener más información, consulte "Configuración de cuotas y reservas de ZFS" en la página 217.

Propiedades de agrupaciones de almacenamiento de ZFS

Versión Solaris 10 10/08: las propiedades de agrupaciones de almacenamiento de ZFS se presentaron en una versión anterior. Esta versión ofrece dos propiedades, `cachefile` y `failmode`.

A continuación, se describen las nuevas propiedades de agrupación de almacenamiento en esta versión:

- La propiedad `cachefile`: esta propiedad controla dónde en la memoria caché se almacena la información sobre la configuración de agrupación. Todas las agrupaciones en la caché se importan automáticamente cuando se inicia el sistema. Sin embargo, los entornos de instalación y administración de clústeres podrían requerir el almacenamiento en caché de esta información en otra ubicación, para impedir la importación automática de las agrupaciones.

  Esta propiedad puede establecerse para que la configuración de la agrupación se guarde en la caché en otra ubicación que luego pueda importarse con el comando `zpool import -c`. Esta propiedad no se utilizará en la mayoría de las configuraciones de ZFS.

  La propiedad `cachefile` no es persistente y no se almacena en el disco. Esta propiedad sustituye la propiedad `temporary` que se usó para indicar que la información de la agrupación no debe guardarse en la caché en versiones anteriores de Solaris.

- La propiedad `failmode`: esta propiedad proporciona el comportamiento de un error grave de agrupación debido a una pérdida de conectividad de dispositivos o al error de todos los dispositivos de la agrupación. La propiedad `failmode` se puede establecer con los valores `wait`, `continue` o `panic`. El valor predeterminado es `wait`, lo que significa que debe volver a conectar el dispositivo, o sustituir un dispositivo anómalo y suprimir el error con el comando `zpool clear`.

  La propiedad `failmode` se establece como otras propiedades configurables de ZFS que se pueden establecer antes o después de crear la agrupación. Por ejemplo:

```
# zpool set failmode=continue tank
# zpool get failmode tank
NAME  PROPERTY  VALUE     SOURCE
tank  failmode  continue  local

# zpool create -o failmode=continue users mirror c0t1d0 c1t1d0
```

  Para ver una descripción de las propiedades de agrupación, consulte la Tabla 4–1.

Mejoras en el historial de comando ZFS (`zpool history`)

Versión Solaris 10 10/08: el comando `zpool history` se ha mejorado para proporcionar las funciones nuevas siguientes:

- Se muestra información de eventos del sistema de archivos ZFS. Por ejemplo:

```
# zpool history
History for 'rpool':
2010-06-23.09:30:12 zpool create -f -o failmode=continue -R /a -m legacy -o
cachefile=/tmp/root/etc/zfs/zpool.cache rpool c1t0d0s0
2010-06-23.09:30:13 zfs set canmount=noauto rpool
2010-06-23.09:30:13 zfs set mountpoint=/rpool rpool
2010-06-23.09:30:13 zfs create -o mountpoint=legacy rpool/ROOT
2010-06-23.09:30:14 zfs create -b 8192 -V 2048m rpool/swap
2010-06-23.09:30:14 zfs create -b 131072 -V 1024m rpool/dump
```

```
2010-06-23.09:30:15 zfs create -o canmount=noauto rpool/ROOT/zfsBE
2010-06-23.09:30:16 zpool set bootfs=rpool/ROOT/zfsBE rpool
2010-06-23.09:30:16 zfs set mountpoint=/ rpool/ROOT/zfsBE
2010-06-23.09:30:16 zfs set canmount=on rpool
2010-06-23.09:30:16 zfs create -o mountpoint=/export rpool/export
2010-06-23.09:30:17 zfs create rpool/export/home
```

- La opción `-l` se puede utilizar para ver el formato completo que incluye el nombre de usuario, el nombre de host y la zona en que se ha efectuado la operación. Por ejemplo:

```
# zpool history -l rpool
History for 'tank':
2010-06-24.13:07:58 zpool create tank mirror c2t2d0 c2t5d0 [user root on neo:global]
2010-06-24.13:08:23 zpool scrub tank [user root on neo:global]
2010-06-24.13:38:42 zpool clear tank [user root on neo:global]
2010-06-29.11:44:18 zfs create tank/home [user root on neo:global]
2010-06-29.13:28:51 zpool clear tank c2t5d0 [user root on neo:global]
2010-06-30.14:07:40 zpool add tank spare c2t1d0 [user root on neo:global
```

- La opción `-i` se puede utilizar para mostrar información sobre eventos internos para tareas de diagnóstico. Por ejemplo:

```
# zpool history -i tank
History for 'tank':
2010-06-24.13:07:58 zpool create tank mirror c2t2d0 c2t5d0
2010-06-24.13:08:23 [internal pool scrub txg:6] func=1 mintxg=0 maxtxg=6
2010-06-24.13:08:23 [internal pool create txg:6] pool spa 22; zfs spa 22; zpl 4; uts neo 5.10 Generic_142909-13 s
2010-06-24.13:08:23 [internal pool scrub done txg:6] complete=1
2010-06-24.13:08:23 zpool scrub tank
2010-06-24.13:38:42 zpool clear tank
2010-06-24.13:38:42 [internal pool scrub txg:69] func=1 mintxg=3 maxtxg=8
2010-06-24.13:38:42 [internal pool scrub done txg:69] complete=1
2010-06-29.11:44:18 [internal create txg:14241] dataset = 34
2010-06-29.11:44:18 zfs create tank/home
2010-06-29.13:28:51 zpool clear tank c2t5d0
2010-06-30.14:07:40 zpool add tank spare c2t1d0
```

Si desea más información sobre el comando `zpool history`, consulte "Solución de problemas con ZFS" en la página 297.

Actualización de sistemas de archivos ZFS (`zfs upgrade`)

Versión Solaris 10 10/08: en esta versión se incluye el comando `zfs upgrade` para aportar a los sistemas de archivos actuales las mejoras en los sistemas de archivos ZFS que haya en el futuro. Las agrupaciones de almacenamiento ZFS cuentan con una función de actualización similar para proporcionar mejoras en las agrupaciones a las agrupaciones de almacenamiento existentes.

Por ejemplo:

```
# zfs upgrade
This system is currently running ZFS filesystem version 3.
```

```
All filesystems are formatted with the current version.
```

Nota – Los sistemas de archivos que se actualizan y los flujos de datos que se crean a partir de dichos sistemas actualizados mediante el comando `zfs send` no quedan accesibles en sistemas que ejecuten versiones de software más antiguas.

Administración delegada de ZFS

Versión Solaris 10 10/08: en esta versión, puede conceder con gran precisión permisos para permitir que los usuarios nonprivileged efectúen taras de administración de ZFS.

Puede usar los comandos `zfs allow` y `zfs unallow` para otorgar y suprimir permisos.

Puede modificar la administración delegada con la propiedad `delegación` de la agrupación. Por ejemplo:

```
# zpool get delegation users
NAME  PROPERTY    VALUE       SOURCE
users  delegation  on          default
# zpool set delegation=off users
# zpool get delegation users
NAME  PROPERTY    VALUE       SOURCE
users  delegation  off         local
```

De forma predeterminada se activa la propiedad `delegation`.

Si desea más información, consulte el Capítulo 9, “Administración delegada de ZFS”, y `zfs`(1M).

Configuración de dispositivos de registro de ZFS independientes

Versión Solaris 10 10/08: ZIL (ZFS Intent Log) se proporciona para satisfacer los requisitos de POSIX de transacciones síncronas. Por ejemplo, las bases de datos precisan con frecuencia que sus transacciones se encuentren en dispositivos de almacenamiento estables al volver de una llamada del sistema. NFS y otras aplicaciones también pueden usar `fsync()` para asegurar la estabilidad de los datos. De forma predeterminada, ZIL se asigna a partir de bloques de la agrupación de almacenamiento principal. En esta versión de Solaris, puede decidir si desea que los bloques ZIL continúen asignándose desde la agrupación de almacenamiento principal o desde un dispositivo de registro independiente. Podría mejorarse el rendimiento utilizando dispositivos independientes en la agrupación de almacenamiento de ZFS, por ejemplo NVRAM o un disco dedicado.

Los dispositivos de registros para ZIL no están relacionados con los archivos del registro de la base de datos.

Puede crear un dispositivo de registro ZFS durante o después de la creación de la agrupación de almacenamiento. Para obtener ejemplos de configuración de dispositivos de registro, consulte "Creación de una agrupación de almacenamiento de ZFS con dispositivos de registro" en la página 76 y "Adición de dispositivos a una agrupación de almacenamiento" en la página 83.

Puede vincular un dispositivo de registro a uno ya creado para crear un dispositivo de registro reflejado. Esta operación es idéntica a la de vincular un dispositivo en una agrupación de almacenamiento sin duplicar.

Para saber si es apropiado configurar un dispositivo de registro de ZFS se deben tener en cuenta los puntos siguientes:

- Cualquier mejora en el rendimiento que haya al implementar un dispositivo de registro independiente está sujeta al tipo dispositivo, la configuración de hardware de la aplicación y la carga de trabajo de la aplicación. Para obtener información preliminar sobre el rendimiento, consulte este blog:

  `http://blogs.sun.com/perrin/entry/slog_blog_or_blogging_on`
- Los dispositivos de registro pueden ser duplicados o sin duplicar, pero RAID-Z no es válido para dispositivos de registro.
- Si no se duplica un dispositivo de registro independiente y falla el dispositivo que contiene el registro, el registro que se almacena vuelve a la agrupación de almacenamiento.
- Los dispositivos se pueden agregar, reemplazar, vincular, desvincular, importar y exportar como parte de la agrupación de almacenamiento de mayor tamaño. Los dispositivos de registro se pueden eliminar a partir de la versión Solaris 10 9/10.
- El tamaño mínimo de un dispositivo de registro es el mismo que el de cada dispositivo en una agrupación, es decir, 64 MB. La cantidad de datos en reproducción que se puede almacenar en un dispositivo de registro es relativamente pequeña. Los bloques de registros se liberan si se ejecuta la transacción de registros (llamada del sistema).
- El tamaño máximo de un dispositivo de registro debe ser aproximadamente la mitad de la memoria física, ya que es la cantidad máxima de datos de reproducción potenciales que se pueden almacenar. Por ejemplo, si un dispositivo tiene una memoria física de 16 GB, el dispositivo de registro debería tener como máximo 8 GB.

Creación de conjuntos de datos de ZFS intermedios

Versión Solaris 10 10/08: la opción `-p` con los comandos `zfs create`, `zfs clone` y `zfs rename` es apta para crear rápidamente un conjunto de datos intermedios no existentes, en el caso de que no existan ya.

En el ejemplo siguiente, se crean conjuntos de datos ZFS (`users/area51`) en la agrupación de almacenamiento `datab`.

```
# zfs list
NAME                        USED  AVAIL  REFER  MOUNTPOINT
datab                       106K  16.5G    18K  /datab
# zfs create -p -o compression=on datab/users/area51
```

Si el conjunto de datos intermedio ya existe durante la operación de creación, ésta se completa satisfactoriamente.

Las propiedades especificadas se aplican al conjunto de datos de destino, no a los del conjunto de datos intermedios. Por ejemplo:

```
# zfs get mountpoint,compression datab/users/area51
NAME                PROPERTY     VALUE                SOURCE
datab/users/area51  mountpoint   /datab/users/area51  default
datab/users/area51  compression  on                   local
```

El conjunto de datos intermedio se crea con el punto de montaje predeterminado. Las propiedades adicionales se desactivan para el conjunto de datos intermedio. Por ejemplo:

```
# zfs get mountpoint,compression datab/users
NAME         PROPERTY     VALUE         SOURCE
datab/users  mountpoint   /datab/users  default
datab/users  compression  off           default
```

Para obtener más información, consulte `zfs(1M)`.

Mejoras en conexión en marcha de ZFS

Versión Solaris 10 10/08: en esta versión, ZFS responde con mayor eficiencia a los dispositivos que se eliminan y puede identificar automáticamente los dispositivos que se insertan.

- Puede sustituir un dispositivo existente por otro equivalente sin tener que usar el comando `zpool replace`.

  La propiedad `autoreplace` controla la sustitución automática de un dispositivo. Si la propiedad se establece en `off`, la sustitución del dispositivo debe iniciarla el administrador mediante el comando `zpool replace`. Si la propiedad se establece en `on`, automáticamente se da formato y se sustituye cualquier dispositivo nuevo que se detecte en esta misma ubicación física como dispositivo que perteneciera anteriormente a la agrupación. El comportamiento predeterminado es `off`.
- El estado `REMOVED` de la agrupación de almacenamiento se proporciona cuando se ha extraído físicamente un dispositivo o repuesto en marcha con el sistema en funcionamiento. Un dispositivo de repuesto en marcha se sustituye por el dispositivo extraído, si lo hay.
- Si un dispositivo se extrae y después se vuelve a insertar, queda conectado. Si el repuesto en marcha se activó al volverse a insertar el dispositivo, el repuesto se extrae cuando termina la operación con conexión.

- La detección automática cuando los dispositivos se extraen o insertan depende del hardware, y quizá no sea compatible en todas las plataformas. Por ejemplo, los dispositivos USB se configuran automáticamente al insertarse. Ahora bien, quizá deba utilizar el comando `cfgadm -c configure` para configurar una unidad SATA.
- Los repuestos en marcha se comprueban periódicamente para asegurarse de que tengan conexión y estén disponibles.

Para obtener más información, consulte `zpool(1M)`.

Cambio de nombre recursivo de instantáneas de ZFS (`zfs rename -r`)

Versión Solaris 10 10/08: se puede cambiar el nombre de manera recursiva de todas las instantáneas de ZFS descendientes con el comando `zfs rename -r`. Por ejemplo:

En primer lugar, se crea una instantánea de un conjunto de sistemas de archivos ZFS.

```
# zfs snapshot -r users/home@today
# zfs list
NAME                     USED  AVAIL  REFER  MOUNTPOINT
users                    216K  16.5G    20K  /users
users/home                76K  16.5G    22K  /users/home
users/home@today           0      -    22K  -
users/home/markm          18K  16.5G    18K  /users/home/markm
users/home/markm@today     0      -    18K  -
users/home/marks          18K  16.5G    18K  /users/home/marks
users/home/marks@today     0      -    18K  -
users/home/neil           18K  16.5G    18K  /users/home/neil
users/home/neil@today      0      -    18K  -
```

A continuación, se cambia el nombre de las instantáneas al día siguiente.

```
# zfs rename -r users/home@today @yesterday
# zfs list
NAME                         USED  AVAIL  REFER  MOUNTPOINT
users                        216K  16.5G    20K  /users
users/home                    76K  16.5G    22K  /users/home
users/home@yesterday           0      -    22K  -
users/home/markm              18K  16.5G    18K  /users/home/markm
users/home/markm@yesterday     0      -    18K  -
users/home/marks              18K  16.5G    18K  /users/home/marks
users/home/marks@yesterday     0      -    18K  -
users/home/neil               18K  16.5G    18K  /users/home/neil
users/home/neil@yesterday      0      -    18K  -
```

Una instantánea es el único tipo de conjunto de datos cuyo nombre puede cambiarse de forma recursiva.

Si desea más información sobre instantáneas, consulte "Información general de instantáneas de ZFS" en la página 225 y esta entrada de blog en la que se describe la creación de instantáneas de recuperación:

http://blogs.sun.com/mmusante/entry/rolling_snapshots_made_easy

Compresión gzip disponible para ZFS

Versión Solaris 10 10/08: en esta versión de Solaris, puede establecer la compresión `gzip` en sistemas de archivos ZFS, además de la compresión `lzjb`. Puede especificar la compresión como `gzip`, o `gzip-`*N*, donde *N* es un valor del 1 al 9. Por ejemplo:

```
# zfs create -o compression=gzip users/home/snapshots
# zfs get compression users/home/snapshots
NAME                  PROPERTY     VALUE            SOURCE
users/home/snapshots  compression  gzip             local
# zfs create -o compression=gzip-9 users/home/oldfiles
# zfs get compression users/home/oldfiles
NAME                  PROPERTY     VALUE           SOURCE
users/home/oldfiles   compression  gzip-9          local
```

Para obtener más información sobre el establecimiento de las propiedades de ZFS, consulte "Configuración de propiedades de ZFS" en la página 206.

Almacenamiento de varias copias de datos de usuarios de ZFS

Versión Solaris 10 10/08: como función de fiabilidad, los metadatos de sistemas de archivos ZFS se guardan automáticamente varias veces en discos distintos, si es posible. Esta función se conoce como *bloques ditto*.

En esta versión de Solaris, puede especificar que también se almacenen varias copias de los datos de usuario por sistema de archivos utilizando el comando `zfs set copies`. Por ejemplo:

```
# zfs set copies=2 users/home
# zfs get copies users/home
NAME        PROPERTY  VALUE       SOURCE
users/home  copies    2           local
```

Los valores disponibles son 1, 2 o 3. El valor predeterminado es 1. Estas copias son adicionales a cualquier redundancia de nivel de grupo, por ejemplo en una configuración RAID-Z o duplicada.

Las ventajas de almacenar varias copias de los datos de usuario ZFS son:

- Mejora la retención de datos al permitir la recuperación de fallos de lectura de bloques irrecuperables, como los fallos de medios (conocidos como *bit rot) para todas las configuraciones ZFS.*
- Proporciona protección de datos, incluso cuando sólo hay disponible un disco.
- Permite seleccionar las directivas de protección de datos por sistema de archivos, más allá de las posibilidades de la agrupación de almacenamiento.

Nota – Según la asignación de los bloques ditto en la agrupación de almacenamiento, varias copias se podrían colocar en un solo disco. Un posible fallo posterior en el disco podría hacer que todos los bloques ditto no estuvieran disponibles.

Los bloques ditto pueden ser útiles cuando de forma involuntaria se crea una agrupación no redundante y se deben establecer directivas de retención de datos.

Si desea obtener una descripción detallada sobre las repercusiones generales en la protección de datos al configurar copias en un sistema con una sola agrupación de un solo disco o una de varios discos, consulte el blog siguiente:

`http://blogs.sun.com/relling/entry/zfs_copies_and_data_protection`

Para obtener más información sobre el establecimiento de las propiedades de ZFS, consulte "Configuración de propiedades de ZFS" en la página 206.

Salida mejorada de `zpool status`

Versión Solaris 10 8/07: puede utilizar el comando `zpool status -v` para que aparezca una lista de archivos con errores continuos. Anteriormente, se usaba el comando `find -inum` para identificar los nombres de archivos de la lista de inodos.

Para obtener más información sobre cómo obtener una lista de archivos con errores continuos, consulte "Reparación de un archivo o directorio dañado" en la página 316.

Mejoras en ZFS y Solaris iSCSI

Solaris 10 8/07: en esta versión de Solaris, puede crear un volumen ZFS como dispositivo de destino iSCSI de Solaris si establece la propiedad `shareiscsi` en el volumen ZFS. Es una forma fácil de configurar rápidamente un destino iSCSI de Solaris. Por ejemplo:

```
# zfs create -V 2g tank/volumes/v2
# zfs set shareiscsi=on tank/volumes/v2
# iscsitadm list target
Target: tank/volumes/v2
    iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
    Connections: 0
```

Una vez creado el destino de iSCSI, puede configurar el iniciador de iSCSI. Para obtener información sobre la configuración de un iniciador de Solaris iSCSI, consulte el Capítulo 14, "Configuring Oracle Solaris iSCSI Targets and Initiators (Tasks)" de *System Administration Guide: Devices and File Systems*.

Para obtener más información sobre cómo administrar un volumen ZFS como destino iSCSI, consulte "Uso de un volumen de ZFS como objetivo iSCSI de Solaris" en la página 283.

Historial de comandos de ZFS (`zpool history`)

Versión Solaris 10 8/07: en esta versión de Solaris, ZFS registra automáticamente comandos `zfs` y `zpool` válidos que modifican la información del estado de la agrupación. Por ejemplo:

```
# zpool history
History for 'newpool':
2007-04-25.11:37:31 zpool create newpool mirror c0t8d0 c0t10d0
2007-04-25.11:37:46 zpool replace newpool c0t10d0 c0t9d0
2007-04-25.11:38:04 zpool attach newpool c0t9d0 c0t11d0
2007-04-25.11:38:09 zfs create newpool/user1
2007-04-25.11:38:15 zfs destroy newpool/user1

History for 'tank':
2007-04-25.11:46:28 zpool create tank mirror c1t0d0 c2t0d0 mirror c3t0d0 c4t0d0
```

Esta función permite al usuario o al personal de asistencia de Oracle identificar los comandos *exactos* de ZFS ejecutados para solucionar una situación de error.

Puede identificar una agrupación de almacenamiento específica con el comando `zpool history`. Por ejemplo:

```
# zpool history newpool
History for 'newpool':
2007-04-25.11:37:31 zpool create newpool mirror c0t8d0 c0t10d0
2007-04-25.11:37:46 zpool replace newpool c0t10d0 c0t9d0
2007-04-25.11:38:04 zpool attach newpool c0t9d0 c0t11d0
2007-04-25.11:38:09 zfs create newpool/user1
2007-04-25.11:38:15 zfs destroy newpool/user1
```

En esta versión de Solaris, el comando `zpool history` no registra *ID_usuario*, *nombre_host* ni *nombre_zona*. Sin embargo, esta información se registra a partir de la versión Solaris 10 10/08. Para obtener más información, consulte "Mejoras en el historial de comando ZFS (`zpool history`)" en la página 32.

Para obtener más información sobre la resolución de los problemas de ZFS, consulte "Solución de problemas con ZFS" en la página 297.

Mejoras en las propiedades de ZFS

Propiedad `xattr` de ZFS

Versión Solaris 10 8/07: puede utilizar la propiedad `xattr` para deshabilitar o habilitar los atributos extendidos de un determinado sistema de archivos ZFS. El valor predeterminado es `on`. Para obtener una descripción de las propiedades de ZFS, consulte "Introducción a las propiedades de ZFS" en la página 189.

Propiedad `canmount` de ZFS

Versión Solaris 10 8/07: la nueva propiedad `canmount` permite especificar si un conjunto de datos se puede montar mediante el comando `zfs mount`. Para obtener más información, consulte "Propiedad canmount" en la página 201.

Propiedades de usuario de ZFS

Versión Solaris 10 8/07: además de las propiedades nativas estándar que pueden exportar estadísticas internas o controlar el comportamiento del sistema de archivos ZFS, éste admite propiedades del usuario. Las propiedades del usuario no repercuten en el comportamiento del sistema de archivos ZFS, pero puede usarlas para anotar información de manera que tenga sentido en su entorno.

Para obtener más información, consulte "Propiedades de usuario de ZFS" en la página 202.

Configuración de propiedades al crear sistemas de archivos ZFS

Versión Solaris 10 8/07: en esta versión de Solaris, puede establecer las propiedades al crear un sistema de archivos, no sólo después de crearlo.

Los ejemplos siguientes ilustran la sintaxis equivalente:

```
# zfs create tank/home
# zfs set mountpoint=/export/zfs tank/home
# zfs set sharenfs=on tank/home
# zfs set compression=on tank/home

# zfs create -o mountpoint=/export/zfs -o sharenfs=on -o compression=on tank/home
```

Visualización de la información de todo el sistema de archivos ZFS

Versión Solaris 10 8/07: en esta versión de Solaris, puede utilizar diversas formas del comando `zfs get` para visualizar información sobre todos los conjuntos de datos si no especifica un conjunto de datos o si especifica `all`. En las versiones anteriores no se podía recuperar toda la información del conjunto de datos con el comando `zfs get`.

Por ejemplo:

```
# zfs get -s local all
tank/home               atime          off                    local
tank/home/bonwick       atime          off                    local
tank/home/marks         quota          50G                    local
```

Nueva opción `zfs receive -F`

Versión Solaris 10 8/07: en esta versión de Solaris, puede utilizar la nueva opción `-F` con el comando `zfs receive` para forzar un restablecimiento del sistema de archivos a la instantánea más reciente antes de ejecutar la recepción. El uso de esta opción podría ser necesario cuando se modifica el sistema de archivos entre el momento en que se realiza la inversión y el momento en que se inicia la recepción.

Para obtener más información, consulte "Recepción de una instantánea ZFS" en la página 236.

Instantáneas de ZFS recurrentes

Versión Solaris 10 11/06: si utiliza el comando `zfs snapshot` para crear una instantánea del sistema de archivos, puede usar la opción `-r` para crear de forma recurrente instantáneas de todos los sistemas de archivos descendientes. Además, puede utilizar la opción `-r` para destruir repetidamente todas las instantáneas descendientes si se destruye una instantánea.

Las instantáneas recurrentes de ZFS se crean con rapidez, como una operación atómica. Las instantáneas se crean todas juntas (a la vez) o no se crea ninguna. La ventaja de esta operación estriba en que los datos de instantánea se toman siempre en un momento coherente, incluso en el caso de sistemas de archivos descendientes.

Para obtener más información, consulte "Creación y destrucción de instantáneas de ZFS" en la página 226.

RAID-Z de paridad doble (`raidz2`)

Versión Solaris 10 11/06: una configuración redundante de RAID-Z puede tener ahora una configuración de paridad sencilla o doble, lo cual significa que se pueden asumir uno o dos errores de dispositivos, sin pérdida de datos. Puede especificar la palabra clave `raidz2` para una configuración de RAID-Z de paridad doble. Otra posibilidad es especificar la palabra clave `raidz` o `raidz1` para una configuración de RAID-Z de paridad sencilla.

Para obtener más información, consulte "Creación de una agrupación de almacenamiento de RAID-Z" en la página 74 o `zpool(1M)`.

Repuestos en marcha para dispositivos de agrupación de almacenamiento de ZFS

Versión Solaris 10 11/06: la función de repuestos en marcha de ZFS permite identificar discos aptos para sustituir un dispositivo con errores en una o varias agrupaciones de

almacenamiento. Al designar un dispositivo como *repuesto en marcha*, si falla un dispositivo activo de la agrupación, el repuesto en marcha lo reemplaza automáticamente. También se puede reemplazar manualmente un dispositivo de un grupo de almacenamiento por un repuesto en marcha.

Para obtener más información, consulte "Designación de repuestos en marcha en la agrupación de almacenamiento" en la página 98 y `zpool(1M)`.

Sustitución de un sistema de archivos ZFS por un clon de ZFS (`zfs promote`)

Versión Solaris 10 11/06: el comando `zfs promote` permite sustituir un sistema de archivos ZFS por un clon de ese sistema de archivos. Es una función útil para ejecutar pruebas en una versión alternativa de un sistema de archivos y después convertirlo en el sistema de archivos activo.

Para obtener más información, consulte "Sustitución de un sistema de archivos ZFS por un clon de ZFS" en la página 233 y `zfs(1M)`.

Actualización de agrupaciones de almacenamiento de ZFS (`zpool upgrade`)

Versión Solaris 10 6/06: puede actualizar las agrupaciones de almacenamiento a una versión más reciente de ZFS para poder utilizar las nuevas funciones mediante el comando `zpool upgrade`. Asimismo, el comando `zpool status` se ha modificado para notificar a los usuarios de que las agrupaciones están ejecutando versiones antiguas de ZFS.

Para obtener más información, consulte "Actualización de agrupaciones de almacenamiento de ZFS" en la página 123 y `zpool(1M)`.

Si desea utilizar la consola de administración de ZFS en un sistema con una agrupación de una versión anterior de Solaris, actualice las agrupaciones antes de usar la consola. Para saber si las agrupaciones se deben actualizar, utilice el comando `zpool status`. Para obtener más información sobre la consola de administración de ZFS, consulte "Administración por Internet de ZFS" en la página 45.

Cambio de nombre en los comandos de restauración y copia de seguridad de ZFS

Versión Solaris 10 6/06: en esta versión de Solaris, se cambia el nombre de los comandos `zfs backup` y `zfs restore` por `zfs send` y `zfs receive` para describir estas funciones de forma más precisa. Estos comandos envían y reciben representaciones de flujos de datos de ZFS.

Para obtener más información sobre estos comandos, consulte "Envío y recepción de datos ZFS" en la página 234.

Recuperación de agrupaciones de almacenamiento destruidas

Versión Solaris 10 6/06: esta versión incluye el comando `zpool import -D`, que permite recuperar agrupaciones destruidas con el comando `zpool destroy`.

Para obtener más información, consulte "Recuperación de agrupaciones de almacenamiento de ZFS destruidas" en la página 121.

ZFS se integra en el administrador de fallos

Versión Solaris 10 6/06: esta versión incluye un motor de diagnóstico de ZFS capaz de diagnosticar errores en dispositivos y agrupaciones, y de generar informes. También se informa de errores de suma de comprobación, E/S, dispositivos y agrupaciones asociados con errores de dispositivos o agrupaciones.

El motor de diagnóstico no incluye análisis predictivos de suma de comprobación ni errores de E/S, ni tampoco acciones proactivas basadas en análisis de errores.

Si se produce un error de ZFS, es posible que vea un mensaje parecido al siguiente:

```
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Wed Jun 30 14:53:39 MDT 2010
PLATFORM: SUNW,Sun-Fire-880, CSN: -, HOSTNAME: neo
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: 504a1188-b270-4ab0-af4e-8a77680576b8
DESC: A ZFS device failed.  Refer to http://sun.com/msg/ZFS-8000-D3 for more information.
AUTO-RESPONSE: No automated response will occur.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.
```

Al revisar la acción recomendada, que será seguir las indicaciones más específicas del comando `zpool status`, podrá identificar el error y solucionarlo rápidamente.

Si desea ver un ejemplo de recuperación de un problema de ZFS del que se ha informado, consulte "Resolución de un dispositivo que no se encuentra" en la página 303.

El comando `zpool clear`

Versión Solaris 10 6/06: esta versión incluye el comando `zpool clear` para borrar los recuentos de errores asociados con un dispositivo o una agrupación. Anteriormente, los

recuentos de errores se suprimían cuando se conectaba el dispositivo de una agrupación mediante el comando `zpool online`. Para obtener más información, consulte "Borrado de errores de dispositivo de agrupación de almacenamiento" en la página 96 y `zpool(1M)`.

Formato compacto NFSv4 de lista de control de acceso (ACL)

Versión Solaris 10 6/06: en esta versión, puede establecer y visualizar listas de control de acceso (ACL) NFSv4 en dos formatos: detallado y formato compacto. Con el comando `chmod` se pueden establecer los tres formatos de lista de control de acceso (ACL). Puede utilizar el `comando ls - V` para visualizar el formato de lista de control de acceso (ACL) comprimido. Puede utilizar el comando `ls -v` para mostrar el formato de lista de control de acceso (ACL) detallado.

Para obtener más información, consulte "Establecimiento y visualización de ACL en archivos ZFS en formato compacto" en la página 262, `chmod(1)` y `ls(1)`.

Herramienta de supervisión del sistema de archivos (`fsstat`)

Versión Solaris 10 6/06: `fsstat`, una nueva herramienta de supervisión del sistema de archivos, informa de las operaciones del sistema de archivos. Se informa de la actividad por punto de montaje o por tipo de sistema de archivos. El ejemplo siguiente muestra la actividad general del sistema de archivos ZFS:

```
$ fsstat zfs
 new  name   name  attr  attr lookup rddir  read read  write write
 file remov  chng   get   set    ops   ops   ops bytes   ops bytes
7.82M 5.92M 2.76M 1.02G 3.32M  5.60G 87.0M  363M 1.86T 20.9M  251G zfs
```

Para obtener más información, consulte `fsstat(1M)`.

Administración por Internet de ZFS

Versión Solaris 10 6/06: la consola de administración de ZFS, una herramienta de administración de ZFS en Internet, permite realizar las siguientes tareas administrativas:

- Crear una agrupación de almacenamiento.
- Agregar capacidad a un grupo.
- Mover (exportar) un grupo de almacenamiento a otro sistema.
- Importar un grupo de almacenamiento previamente exportado para que quede disponible en otro sistema.

- Ver información sobre grupos de almacenamiento.
- Crear un sistema de archivos.
- Crear un volumen.
- Crear una instantánea de un sistema de archivos o un volumen.
- Restaurar un sistema de archivos a una instantánea anterior.

Puede acceder a la consola de administración de ZFS mediante un navegador Web seguro en:

```
https://system-name:6789/zfs
```

Si escribe la dirección URL pertinente y no puede acceder a la consola de administración de ZFS, es posible que el servidor no se inicie. Para iniciarlo, ejecute el siguiente comando:

```
# /usr/sbin/smcwebserver start
```

Si desea que el servidor se ejecute automáticamente al arrancar el sistema, ejecute el siguiente comando:

```
# /usr/sbin/smcwebserver enable
```

Nota – No se puede utilizar Solaris Management Console (`smc`) para administrar sistemas de archivos o grupos de almacenamiento ZFS.

Definición de ZFS

ZFS es un nuevo y revolucionario sistema de archivos que aporta una forma totalmente distinta de administrar sistemas de archivos, con funciones y ventajas que no hay en ningún otro sistema de archivos actual. ZFS es sólido, escalable y fácil de administrar.

Almacenamiento en agrupaciones de ZFS

ZFS se basa en el concepto de *grupos de almacenamiento* para administrar el almacenamiento físico. Desde siempre, los sistemas de archivos se estructuran a partir de un solo dispositivo físico. Para poder ocuparse de varios dispositivos y ofrecer redundancia de datos, se incorporó el concepto del *administrador de volúmenes*, con el fin de ofrecer una representación de un único dispositivo y evitar que los sistemas de archivos tuvieran que modificarse para aprovechar las ventajas de varios dispositivos. Este diseño significaba otro nivel de complejidad y obstaculizaba determinados avances en los sistemas de archivos, al carecer de control sobre la ubicación física de los datos en los volúmenes virtualizados

ZFS elimina del todo la administración de volúmenes. En vez de tener que crear volúmenes virtualizados, ZFS agrega dispositivos a una agrupación de almacenamiento. La agrupación de almacenamiento describe las características físicas del almacenamiento (organización del

dispositivo, redundancia de datos, etc.) y actúa como almacén de datos arbitrario en el que se pueden crear sistemas de archivos. Los sistemas de archivos ya se limitan a dispositivos individuales y les permite compartir espacio en el disco con todos los sistemas de archivos de la agrupación. Ya no es necesario predeterminar el tamaño de un sistema de archivos, ya que el tamaño de los sistemas de archivos crece automáticamente en el espacio asignado a la agrupación de almacenamiento. Al incorporar un nuevo almacenamiento, todos los sistemas de archivos de la agrupación pueden usar de inmediato el espacio en el disco adicional sin procesos complementarios. En muchos sentidos, la agrupación de almacenamiento funciona del mismo modo que un sistema de memoria virtual: si se agrega al sistema un módulo de memoria DIMM, el sistema operativo no obliga a ejecutar comandos para configurar la memoria y asignarla a los procesos individuales. Todos los procesos del sistema utilizan automáticamente la memoria adicional.

Semántica transaccional

ZFS es un sistema de archivos transaccional. Ello significa que el estado del sistema de archivos siempre es coherente en el disco. Los sistemas de archivos tradicionales sobrescriben datos in situ. Esto significa que, si el equipo se queda sin alimentación (por ejemplo, entre el momento en que un bloque de datos se asigna y cuando se vincula a un directorio), el sistema de archivos se queda en un estado incoherente. En el pasado, este problema se solucionaba mediante el comando `fsck`. Este comando verificaba el estado del sistema de archivos e intentaba reparar cualquier incoherencia durante el proceso. Este problema de sistemas de archivos incoherentes daba muchos quebraderos de cabeza a los administradores y el comando `fsck` nunca garantizaba la solución a todos los problemas. Posteriormente, los sistemas de archivos han incorporado el concepto de *registro de diario*. *El registro de diario guarda las acciones en un diario aparte, el cual se puede volver a reproducir con seguridad si el sistema se bloquea.* Este proceso supone cargas innecesarias, porque los datos se deben escribir dos veces y a menudo provoca una nueva fuente de problemas (como no poder volver a reproducir correctamente el registro de diario).

Con un sistema de archivos transaccional, los datos se administran mediante la semántica *copy on write*. Los datos nunca se sobrescriben y ninguna secuencia de operaciones se confirma o ignora por completo. Este mecanismo hace que el sistema de archivos nunca pueda dañarse por una interrupción imprevista de la alimentación o un bloqueo del sistema. Aunque pueden perderse fragmentos de datos escritos más recientemente, el propio sistema de archivos siempre será coherente. Asimismo, siempre se garantiza que los datos sincrónicos (escritos mediante el indicador `O_DSYNC`) se escriban antes de la devolución, por lo que nunca se pierden.

Datos de reparación automática y sumas de comprobación

En ZFS se verifican todos los datos y metadatos mediante un algoritmo de suma de comprobación seleccionable por el usuario. Los sistemas de archivos tradicionales con suma de comprobación la efectúan por bloques obligatoriamente debido a la capa de administración de volúmenes y la disposición del sistema de archivos tradicional. El diseño tradicional significa que algunos errores, como la escritura de un bloque completo en una ubicación incorrecta, pueden hacer que los datos no sean correctos, pero no producen errores de suma de comprobación. Las sumas de comprobación de ZFS se almacenan de forma que estos errores se detecten y haya una recuperación eficaz. La suma de comprobación y recuperación de datos se efectúan en la capa del sistema de archivos, y son transparentes para las aplicaciones.

Asimismo, ZFS ofrece soluciones para la reparación automática de datos. ZFS admite agrupaciones de almacenamiento con diversos niveles de redundancia de datos. Si se detecta un bloque de datos incorrectos, ZFS recupera los datos correctos de otra copia redundante y repara los datos incorrectos al sustituirlos por una copia correcta.

Escalabilidad incomparable

Un elemento de diseño clave en el sistema de archivos ZFS es la escalabilidad. El sistema de archivos es de 128 bits y permite 256 trillones de zettabytes de almacenamiento. Todos los metadatos se asignan de forma dinámica, con lo que no hace falta asignar previamente inodos ni limitar la escalabilidad del sistema de archivos cuando se crea. Todos los algoritmos se han escrito teniendo en cuenta la escalabilidad. Los directorios pueden tener hasta 2^{48} (256 billones) de entradas; no existe un límite para el número de sistemas de archivos o de archivos que puede haber en un sistema de archivos.

Instantáneas de ZFS

Una *instantánea* es una copia de sólo lectura de un sistema de archivos o volumen. Las instantáneas se crean rápida y fácilmente. Inicialmente, las instantáneas no consumen espacio adicional en el disco dentro de la agrupación.

Como los datos de un conjunto de datos activo cambian, la instantánea consume espacio en el disco al seguir haciendo referencia a los datos antiguos. Como resultado, la instantánea impide que los datos vuelvan a pasar a la agrupación.

Administración simplificada

Uno de los aspectos más destacados de ZFS es su modelo de administración muy simplificado. Mediante un sistema de archivos con distribución jerárquica, herencia de propiedades y administración automática de puntos de montaje y semántica share de NFS, ZFS facilita la

creación y administración de sistemas de archivos sin tener que usar varios comandos ni editar archivos de configuración. Con un solo comando puede establecer fácilmente cuotas o reservas, activar o desactivar la compresión, o administrar puntos de montaje para diversos sistemas de archivos. Puede examinar o sustituir dispositivos sin aprender un conjunto independiente de comandos de administrador de volúmenes. Puede enviar y recibir flujos de instantáneas del sistema de archivos .

ZFS administra los sistemas de archivos a través de una jerarquía que permite la administración simplificada de propiedades como cuotas, reservas, compresión y puntos de montaje. En este modelo, los sistemas de archivos se convierten en el punto central de control. Los sistemas de archivos son muy sencillos (equivalen a un nuevo directorio), por lo que se recomienda crear un sistema de archivos para cada usuario, proyecto, espacio de trabajo, etc. Este diseño permite definir los puntos de administración de forma detallada.

Terminología de ZFS

Esta sección describe la terminología básica utilizada en este manual:

entorno de inicio alternativo	Entorno de inicio que se ha creado con el comando `lucreate` y posiblemente se ha actualizado mediante el comando `luupgrade`, pero que no es ni el entorno de inicio activo ni el primario. El entorno de inicio alternativo puede convertirse en el entorno de inicio primario con la ejecución del comando `luactivate`.
suma de comprobación	Cifrado de 256 bits de los datos en un bloque del sistema de archivos. La suma de comprobación puede ir de la rápida y sencilla fletcher4 (valor predeterminado) a cifrados criptográficamente complejos como SHA256.
clon	Sistema de archivos cuyo contenido inicial es idéntico al de una instantánea. Para obtener más información sobre clones, consulte "Información general sobre clones de ZFS" en la página 232.
conjunto de datos	Nombre genérico de las entidades ZFS siguientes: clones, sistemas de archivos, instantáneas y volúmenes. Cada conjunto de datos se identifica mediante un nombre exclusivo en el espacio de nombres de ZFS. Los conjuntos de datos se identifican mediante el formato siguiente: *agrupación*/*ruta*[@*instantánea*]

instantánea	Identifica el nombre de la agrupación de almacenamiento que contiene el conjunto de datos
ruta	Nombre de ruta delimitado por barras para el componente del conjunto de datos
instantánea	Componente opcional que identifica una instantánea de un conjunto de datos

Para obtener más información sobre conjuntos de datos, consulte el Capítulo 6, "Administrar sistemas de archivos ZFS de Oracle Solaris".

sistema de archivos — Conjunto de datos de ZFS del tipo `filesystem` que se monta en el espacio de nombre del sistema estándar y se comporta igual que otros sistemas de archivos.

Para obtener más información sobre sistemas de archivos, consulte el Capítulo 6, "Administrar sistemas de archivos ZFS de Oracle Solaris".

duplicación — Dispositivo virtual que almacena copias idénticas de datos en dos discos o más. Si falla cualquier disco de una duplicación, cualquier otro disco de esa duplicación puede proporcionar los mismos datos.

agrupación — Conjunto lógico de dispositivos que describe la disposición y las características físicas del almacenamiento disponible. El espacio en el disco para conjuntos de datos que se asigna a partir de una agrupación.

For more information about storage pools, see Capítulo 4, "Administración de agrupaciones de almacenamiento de Oracle Solaris ZFS".

entorno de inicio principal — Entorno de inicio que utiliza el comando `lucreate` para crear el entorno de inicio alternativo. De forma predeterminada, el entorno de inicio principal es el entorno de inicio actual. Este valor predeterminado se puede cambiar con la opción `lucreate -s`.

RAID-Z — Dispositivo virtual que almacena datos y la paridad en varios discos. Para obtener más información sobre RAID-Z, consulte "Configuración de agrupaciones de almacenamiento RAID-Z" en la página 70.

actualización de duplicación	El proceso de transferir datos de un dispositivo a otro se denomina *actualización de duplicación*. Por ejemplo, si un dispositivo de duplicación se sustituye o se desconecta, los datos actualizados del dispositivo de duplicación se copian en el dispositivo de duplicación recién restaurado. Este proceso se denomina *resincronización de duplicación* en productos tradicionales de administración de volúmenes. Si desea más información sobre la actualización de duplicación ZFS, consulte "Visualización del estado de la actualización de duplicación de datos" en la página 313.
instantánea	Imagen de sólo lectura de un sistema de archivos o volumen de un momento determinado. Para obtener más información sobre instantáneas, consulte "Información general de instantáneas de ZFS" en la página 225.
dispositivo virtual	Dispositivo lógico de un grupo que puede ser un dispositivo físico, un archivo o un conjunto de dispositivos. Si desea más información sobre dispositivos virtuales, consulte "Visualización de información de dispositivos virtuales de agrupaciones de almacenamiento" en la página 77.
volumen	Un conjunto de datos que representa un dispositivo de bloques. Por ejemplo, puede crear un volumen de ZFS como dispositivo de intercambio. Para obtener más información sobre volúmenes de ZFS, consulte "Volúmenes de ZFS" en la página 281.

Requisitos de asignación de nombres de componentes de ZFS

Cada componente de ZFS (por ejemplo, conjunto de datos y agrupación) debe recibir un nombre según las reglas siguientes:

- Cada componente sólo puede contener caracteres alfanuméricos, además de los cuatro caracteres especiales siguientes:
  - Guión bajo (_)
  - Guión (-)
  - Dos puntos (:)
  - Punto (.)

- Los nombres de las agrupaciones deben comenzar con una letra, pero teniendo en cuenta las limitaciones siguientes:
  - No se permite la secuencia de inicio `c[0-9]`.
  - El nombre `log` está reservado.
  - No se permiten los nombres que comiencen por `mirror`, `raidz` , `raidz1`, `raidz2`, `raidz3` , o `spare` porque dichos nombres están reservados.
  - Los nombres de las agrupaciones de datos no pueden contener un signo porcentual (`%`).
- Los nombres de los conjuntos de datos deben comenzar por un carácter alfanumérico.
- Los nombres de los conjuntos de datos no pueden contener un signo porcentual (`%`).

Además, no se permiten los componentes vacíos.

◆ ◆ ◆ CAPÍTULO 2

Procedimientos iniciales con Oracle Solaris ZFS

Este capítulo proporciona instrucciones paso a paso para definir una configuración básica de Oracle Solaris ZFS. Al terminar este capítulo, habrá adquirido nociones básicas sobre el funcionamiento de los comandos de ZFS, y debería ser capaz de crear sistemas de archivos y una agrupación sencilla. Este capítulo no profundiza en el contenido. Para obtener información más detallada, consulte los capítulos siguientes.

Este capítulo se divide en las secciones siguientes:

Recomendaciones y requisitos de software y hardware para ZFS

Antes de utilizar el software de ZFS, revise los requisitos y las recomendaciones de software y hardware siguientes:

- Use un sistema SPARC o x86 que ejecute la versión Solaris 10 6/06 o posterior.
- Una agrupación de almacenamiento necesita como mínimo 64 MB de espacio en el disco. El tamaño de disco mínimo es 128 MB.
- La cantidad mínima de memoria necesaria para instalar un sistema Solaris es 768 MB. Pero para un buen rendimiento de ZFS, utilice al menos un GB de memoria.
- Si crea una configuración de disco reflejada, utilice varios controladores.

Creación de un sistema de archivos ZFS básico

Se ha intentado diseñar la administración de ZFS con la máxima sencillez posible. Entre los objetivos del diseño está la reducción del número de comandos necesarios para crear un sistema de archivos utilizable. Por ejemplo, al crear una agrupación, se crea un sistema de archivos ZFS y se monta automáticamente.

El ejemplo siguiente ilustra la manera de crear una agrupación de almacenamiento reflejado denominado `tank` y un sistema de archivos ZFS denominado `tank` en un comando. Suponga que se pueden utilizar todos los discos `/dev/dsk/c1t0d0` y `/dev/dsk/c2t0d0`.

```
# zpool create tank mirror c1t0d0 c2t0d0
```

Para obtener más información sobre configuraciones de agrupaciones ZFS redundantes, consulte "Funciones de repetición de una agrupación de almacenamiento de ZFS" en la página 69.

El nuevo sistema de archivos ZFS, `tank`, puede usar tanto espacio como necesite y se monta automáticamente en `/tank`.

```
# mkfile 100m /tank/foo
# df -h /tank
Filesystem             size   used  avail capacity  Mounted on
tank                   80G   100M    80G     1%    /tank
```

Quizá desee crear sistemas de archivos adicionales en una agrupación. Los sistemas de archivos ofrecen puntos que permiten administrar distintos conjuntos de datos en la misma agrupación.

El ejemplo siguiente ilustra la manera de crear un sistema de archivos denominado `fs` en la agrupación de almacenamiento `tank`.

```
# zfs create tank/fs
```

El nuevo sistema de archivos ZFS, `tank/fs`, puede utilizar la cantidad de espacio en el disco que necesite y se monta automáticamente en `/tank/fs`.

```
# mkfile 100m /tank/fs/foo
# df -h /tank/fs
Filesystem             size   used  avail capacity  Mounted on
tank/fs                 80G   100M    80G     1%    /tank/fs
```

Normalmente, el objetivo es crear y organizar una jerarquía de sistemas de archivos que se ajuste a los requisitos de su organización. Para obtener más información sobre cómo crear jerarquías de sistemas de archivos ZFS, consulte "Creación de una jerarquía para el sistema de archivos ZFS" en la página 57.

Creación de una agrupación de almacenamiento de ZFS

El ejemplo anterior es una muestra de la sencillez de ZFS. El resto de este capítulo expone un ejemplo más completo y similar a la situación de su entorno. Las primeras tareas son establecer los requisitos de almacenamiento y crear una agrupación de almacenamiento. La agrupación describe las características físicas del almacenamiento y se deben crear antes que un sistema de archivos.

▼ Identificación de los requisitos de la agrupación de almacenamiento de ZFS

1 Averigüe qué dispositivos están disponibles para la agrupación de almacenamiento.

Antes de crear una agrupación de almacenamiento, debe establecer los dispositivos que almacenarán los datos. Deben ser discos de al menos 128 MB y no los deben utilizar otros componentes del sistema operativo. Los dispositivos pueden ser segmentos de disco al que se ha dado formato previamente, o discos completos a los que ZFS da formato como un único segmento grande.

En el ejemplo de almacenamiento de "Cómo crear una agrupación de almacenamiento de ZFS" en la página 56, suponga que se pueden utilizar los discos `/dev/dsk/c2t0d0` y `/dev/dsk/c0t1d0` completos.

Para obtener más información sobre los discos y cómo se utilizan y etiquetan, consulte "Uso de discos en una agrupación de almacenamiento de ZFS" en la página 66.

2 Seleccione la repetición de datos.

ZFS admite diversos tipos de repetición de datos; esto determina los tipos de errores de hardware que puede soportar la agrupación. ZFS admite configuraciones no redundantes (repartidas en bandas), así como reflejo y RAID-Z (una variación de RAID-5).

En el ejemplo de almacenamiento de "Cómo crear una agrupación de almacenamiento de ZFS" en la página 56, se utiliza el reflejo básico de dos discos disponibles.

Si desea más información sobre las características de repetición de ZFS, consulte "Funciones de repetición de una agrupación de almacenamiento de ZFS" en la página 69.

▼ Cómo crear una agrupación de almacenamiento de ZFS

1 **Adquiera el perfil de usuario root o asuma una función equivalente con el perfil adecuado de derechos de ZFS.**

Para obtener más información sobre los perfiles de derechos de ZFS, consulte "Perfiles de derechos de ZFS" en la página 291.

2 **Elija un nombre para la agrupación de almacenamiento.**

El nombre de agrupación sirve para identificar la agrupación de almacenamiento cuando se utilizan los comandos `zpool` y `zfs`. La mayoría de los sistemas sólo necesitan una agrupación, de manera que puede elegir el nombre que prefiera, siempre y cuando cumpla los requisitos de asignación de nombres especificados en "Requisitos de asignación de nombres de componentes de ZFS" en la página 51.

3 **Cree la agrupación.**

Por ejemplo, el siguiente comando crea una agrupación reflejada denominada `tank`:

```
# zpool create tank mirror c1t0d0 c2t0d0
```

Si uno o más dispositivos contienen otro sistema de archivos o se están utilizando, el comando no puede crear la agrupación.

Para obtener más información sobre cómo crear agrupaciones de almacenamiento, consulte "Creación de una agrupación de almacenamiento de ZFS" en la página 72. Para obtener más información sobre cómo establecer el uso de dispositivos, consulte "Detección de dispositivos en uso" en la página 79.

4 **Examine los resultados.**

Puede determinar si la agrupación se ha creado correctamente mediante el comando `zpool list`.

```
# zpool list
NAME                    SIZE    ALLOC   FREE    CAP  HEALTH     ALTROOT
tank                     80G    137K     80G     0%  ONLINE     -
```

Para obtener más información sobre cómo ver el estado de las agrupaciones, consulte "Consulta del estado de una agrupación de almacenamiento de ZFS" en la página 107.

Creación de una jerarquía para el sistema de archivos ZFS

Después de crear una agrupación de almacenamiento para almacenar los datos, puede crear la jerarquía del sistema de archivos. Las jerarquías son mecanismos sencillos pero potentes para organizar la información. También resultan muy familiares a quienes hayan utilizado un sistema de archivos.

ZFS permite que los sistemas de archivos se organicen en jerarquías, donde cada sistema de archivos tiene un solo superior. La raíz de la jerarquía siempre es el nombre de la agrupación. ZFS integra esta jerarquía mediante la admisión de herencia de propiedades, de manera que las propiedades habituales se puedan configurar rápida y fácilmente en todos los árboles de los sistemas de archivos.

▼ Cómo establecer la jerarquía del sistema de archivos ZFS

1 Elija la granularidad del sistema de archivos.

Los sistemas de archivos ZFS son el punto central de administración. Son ligeros y se pueden crear fácilmente. Un modelo perfectamente válido es un sistema de archivos por usuario o proyecto, ya que posibilita propiedades, instantáneas y copias de seguridad que se controlan por usuario o por proyecto.

Se crean dos sistemas de archivos ZFS, `bonwick` y `billm`, en "Creación de sistemas de archivos ZFS" en la página 58.

Para obtener más información sobre la administración de sistemas de archivos, consulte el Capítulo 6, "Administrar sistemas de archivos ZFS de Oracle Solaris".

2 Agrupe sistemas de archivos similares.

ZFS permite que los sistemas de archivos se organicen en jerarquías, de modo que se puedan agrupar los sistemas de archivos similares. Este modelo ofrece un punto central de administración para controlar propiedades y administrar sistemas de archivos. Los sistemas de archivos similares se deben crear con un nombre común.

En el ejemplo de "Creación de sistemas de archivos ZFS" en la página 58, los dos sistemas de archivos se ubican en un sistema de archivos denominado `home`.

3 Seleccione las propiedades del sistema de archivos.

La mayoría de las características del sistema de archivos se controlan mediante propiedades. Dichas propiedades controlan diversos comportamientos, por ejemplo la ubicación donde se montan los sistemas de archivos, su manera de compartirse, si utilizan compresión y si se ejecuta alguna cuota.

En el ejemplo de "Creación de sistemas de archivos ZFS" en la página 58, todos los directorios de inicio se montan en `/export/zfs/` *usuario*, se comparten mediante NFS y se habilita la compresión. Además se aplica una cuota de 10 GB en `bonwick`.

Para obtener más información sobre propiedades, consulte "Introducción a las propiedades de ZFS" en la página 189.

▼ Creación de sistemas de archivos ZFS

1 Adquiera el perfil de usuario root o asuma una función equivalente con el perfil adecuado de derechos de ZFS.

Para obtener más información sobre los perfiles de derechos de ZFS, consulte "Perfiles de derechos de ZFS" en la página 291.

2 Cree la jerarquía que necesite.

En este ejemplo, se crea un sistema de archivos que actúa como contenedor de determinados sistemas de archivos.

```
# zfs create tank/home
```

3 Configure las propiedades heredadas.

Después de establecer la jerarquía del sistema de archivos, configure las propiedades que deben compartir todos los usuarios:

```
# zfs set mountpoint=/export/zfs tank/home
# zfs set sharenfs=on tank/home
# zfs set compression=on tank/home
# zfs get compression tank/home
NAME             PROPERTY       VALUE                      SOURCE
tank/home        compression    on                         local
```

Las propiedades del sistema de archivos pueden establecerse al crear dicho sistema de archivos. Por ejemplo:

```
# zfs create -o mountpoint=/export/zfs -o sharenfs=on -o compression=on tank/home
```

Para obtener más información sobre propiedades y herencia de propiedades, consulte "Introducción a las propiedades de ZFS" en la página 189.

A continuación, los sistemas de archivos se agrupan en el sistema de archivos `home` en la agrupación `tank`.

4 Cree los sistemas de archivos.

Puede que los sistemas de archivos se hayan creado y que las propiedades se hayan cambiado en el nivel de `home`. Todas las propiedades se pueden cambiar dinámicamente mientras se utilizan los sistemas de archivos.

```
# zfs create tank/home/bonwick
# zfs create tank/home/billm
```

Estos sistemas de archivos heredan los valores de propiedades de sus superiores, de modo que se montan automáticamente en `/export/zfs/` *usuario* y se comparten con NFS. No hace falta editar el archivo `/etc/vfstab` ni `/etc/dfs/dfstab`.

Para obtener más información sobre cómo crear sistemas de archivos, consulte "Creación de un sistema de archivos ZFS" en la página 186.

Para obtener más información sobre el montaje y la compartición de sistemas de archivos, consulte "Montaje y compartición de sistemas de archivos ZFS" en la página 211.

5 Configure las propiedades específicas del sistema de archivos.

En este ejemplo, se asigna una cuota de 10 GB al usuario `bonwick`. Esta propiedad establece un límite en la cantidad de espacio que puede consumir, sea cual sea el espacio disponible en la agrupación.

```
# zfs set quota=10G tank/home/bonwick
```

6 Examine los resultados.

Consulte la información disponible sobre el sistema de archivos mediante el comando `zfs list`:

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
tank                  92.0K  67.0G   9.5K  /tank
tank/home             24.0K  67.0G     8K  /export/zfs
tank/home/billm          8K  67.0G     8K  /export/zfs/billm
tank/home/bonwick        8K  10.0G     8K  /export/zfs/bonwick
```

El usuario `bonwick` sólo tiene disponible un espacio de 10 GB, mientras que el usuario `billm` puede utilizar toda la agrupación (67 GB).

Para obtener más información sobre cómo ver el estado del sistema de archivos, consulte "Consulta de información del sistema de archivos ZFS" en la página 203.

Para obtener más información sobre cómo se utiliza y calcula el espacio en el disco, consulte "Cálculo del espacio de ZFS" en la página 62.

◆ ◆ ◆ CAPÍTULO 3

Oracle Solaris ZFS y sistemas de archivos tradicionales

En este capítulo se explican algunas diferencias destacadas entre Oracle Solaris ZFS y los sistemas de archivos tradicionales. Conocer estas diferencias fundamentales solventará dudas al usar herramientas tradicionales junto con ZFS.

Este capítulo se divide en las secciones siguientes:

Granularidad de sistemas de archivos ZFS

Desde siempre, los sistemas de archivos se han limitado a un dispositivo y, por lo tanto, al tamaño de dicho dispositivo. Crear y volver a crear sistemas de archivos tradicionales debido a las limitaciones de tamaño requiere mucho tiempo y llega a ser complicado. Los productos tradicionales de administración de volúmenes ayudan a llevar a cabo este proceso.

Como los sistemas de archivos ZFS no se limitan a determinados dispositivos, se pueden crear con rapidez y facilidad, de forma parecida a la creación de directorios. Los sistemas de archivos ZFS aumentan automáticamente en el espacio asignado a la agrupación de almacenamiento en la que residen.

En vez de crear un sistema de archivos, por ejemplo `/export/home`, para administrar numerosos subdirectorios de usuarios, puede crear un sistema de archivos por usuario. Puede configurar y administrar fácilmente un gran número de sistemas de archivos aplicando propiedades que pueden heredar los sistemas de archivos descendientes dentro de la jerarquía.

Consulte "Creación de una jerarquía para el sistema de archivos ZFS" en la página 57 para ver un ejemplo de creación de una jerarquía de sistema de archivos.

Cálculo del espacio de ZFS

ZFS se basa en el concepto de almacenamiento en agrupaciones. A diferencia de los sistemas de archivos habituales, asignados al almacenamiento físico, todos los sistemas de archivos ZFS de una agrupación comparten el espacio de almacenamiento de la agrupación. Por lo tanto, el espacio disponible en el disco notificado por utilidades como `df` puede llegar a cambiar aunque el sistema de archivos no esté activo, debido a que otros sistemas de archivos de la agrupación consumen o liberan espacio.

El tamaño máximo de los sistemas de archivos se puede restringir mediante cuotas. Para obtener información sobre las cuotas, consulte "Establecimiento de cuotas en sistemas de archivos ZFS" en la página 218. Se puede garantizar una cantidad determinada de espacio en el disco para un sistema de archivos mediante reserva. Para obtener información acerca de las reservas, consulte "Establecimiento de reservas en sistemas de archivos ZFS" en la página 221. Este modelo es muy similar al de NFS, en el que varios directorios se montan desde el mismo sistema de archivos (`/home`).

Todos los metadatos de ZFS se asignan de forma dinámica. Casi todos los demás sistemas de archivos preasignan gran parte de sus metadatos. Al crearse el sistema de archivos, el resultado es un coste inmediato de asignación de espacio para estos metadatos. También significa que está predefinida la cantidad de archivos que admiten los sistemas de archivos. Como ZFS asigna sus metadatos conforme los necesita, no precisa asignación inicial de espacio y la cantidad de archivos que puede admitir está sólo en función del espacio disponible en el disco. La salida del comando `df -g` no significa lo mismo en ZFS que en otros sistemas. El valor de `total files` (total de archivos) que aparece es sólo un cálculo basado en la cantidad de almacenamiento disponible en la agrupación.

ZFS es un sistema de archivos transaccional. Casi todas las modificaciones de sistemas de archivos se incluyen en grupos de transacciones y se envían al disco de manera asíncrona. Hasta que no se envían al disco, se denominan *cambios pendientes*. La cantidad de espacio en el disco utilizado, disponible y que hace referencia a un archivo o sistema de archivos no tiene en cuenta los cambios pendientes. Los cambios pendientes suelen calcularse en pocos segundos. El hecho de enviar un cambio al disco mediante `fsync(3c)` o `O_SYNC` no garantiza necesariamente la actualización inmediata del espacio que se utiliza en el disco.

Para obtener información más detallada sobre el consumo de espacio en el disco de ZFS notificado por los comandos `du` y `df`, consulte:

`http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq/#whydusize`

Comportamiento de falta de espacio

En ZFS, las instantáneas se crean sin dificultad ni coste alguno. Las instantáneas son comunes en casi todos los entornos de ZFS. Para obtener información sobre instantáneas de ZFS, consulte el Capítulo 7, "Uso de clones e instantáneas de Oracle Solaris ZFS".

La presencia de instantáneas puede producir comportamientos imprevistos al intentar liberar espacio en el disco. En general, con los permisos pertinentes, es posible eliminar archivos de un sistema de archivos lleno y disponer así de más espacio en el disco en el sistema de archivos. No obstante, si el archivo que se va a eliminar existe en una instantánea del sistema de archivos, suprimirlo no proporcionará más espacio libre. Se sigue haciendo referencia a los bloques utilizados por el archivo desde la instantánea.

Como consecuencia, eliminar un archivo puede suponer más consumo del espacio en el disco, ya que para reflejar el nuevo estado del espacio de nombre se debe crear una versión nueva del directorio. Este comportamiento significa que al intentar eliminar un archivo se puede generar un error `ENOSPC` o `EDQUOT` imprevisto.

Montaje de sistemas de archivos ZFS

ZFS reduce la complejidad y facilita la administración. Por ejemplo, en los sistemas de archivos tradicionales debe editar el archivo `/etc/vfstab` cada vez que agregue un sistema de archivos nuevo. ZFS ha suprimido este requisito al montar y desmontar automáticamente los sistemas de archivos en función de las propiedades del conjunto de datos. Las entradas de ZFS no hace falta administrarlas en el archivo `/etc/vfstab`.

Para obtener más información sobre cómo montar y compartir sistemas de archivos ZFS, consulte "Montaje y compartición de sistemas de archivos ZFS" en la página 211.

Administración tradicional de volúmenes

Como se explica en "Almacenamiento en agrupaciones de ZFS" en la página 46, con ZFS no se necesita un administrador de volúmenes aparte. ZFS funciona en dispositivos básicos, lo que permite crear una agrupación de almacenamiento a base de volúmenes lógicos, ya sea de software o hardware. No se recomienda esta configuración, puesto que el funcionamiento óptimo de ZFS se da con dispositivos físicos básicos. El uso de volúmenes lógicos puede perjudicar el rendimiento, la fiabilidad o ambas cosas, y se debe evitar.

Nuevo modelo de LCA de Solaris

Las versiones anteriores del sistema operativo Solaris admitían una implementación de LCA que se basaba sobre todo en la especificación LCA de borrador POSIX. Las ACL basadas en el borrador POSIX se utilizan para proteger los archivos UFS. Se emplea un nuevo modelo Solaris ACL basado en la especificación NFSv4 para proteger archivos ZFS.

A continuación se exponen las diferencias principales del nuevo modelo Solaris ACL:

- El modelo se basa en la especificación de NFSv4 y se parece a las ACL del tipo NT.

- Este modelo ofrece un conjunto mucho más granular de privilegios de acceso.
- Las listas ACL se definen y visualizan con los comandos `chmod` e `ls`, en lugar de los comandos `setfacl` y `getfacl`.
- Semántica heredada mucho más rica para establecer la forma en que se aplican privilegios de acceso del directorio a los subdirectorios, y así sucesivamente.

Para obtener más información sobre el uso de las ACL con archivo ZFS, consulte el Capítulo 8, "Uso de listas ACL para proteger archivos ZFS".

◆ ◆ ◆ CAPÍTULO 4

Administración de agrupaciones de almacenamiento de Oracle Solaris ZFS

Este capítulo describe cómo crear y administrar agrupaciones de almacenamiento en Oracle Solaris ZFS.

Este capítulo se divide en las secciones siguientes:

Componentes de una agrupación de almacenamiento de ZFS

Las secciones siguientes ofrecen información detallada sobre estos componentes de agrupación de almacenamiento:

Uso de discos en una agrupación de almacenamiento de ZFS

El elemento más básico de una agrupación de almacenamiento es el almacenamiento físico. El almacenamiento físico puede ser cualquier dispositivo de bloque de al menos 128 MB. En general, este dispositivo es una unidad de disco duro visible en el sistema, en el directorio `/dev/dsk`.

Un dispositivo de almacenamiento puede ser todo un disco (`c1t0d0`) o un determinado segmento (`c0t0d0s7`). Se recomienda utilizar un disco entero, para lo cual no hace falta dar ningún formato especial al disco. ZFS da formato al disco mediante la etiqueta EFI para que contenga un solo segmento grande. Si se utiliza de este modo, la tabla de partición que aparece junto al comando `format` tiene un aspecto similar al siguiente:

```
Current partition table (original):
Total disk sectors available: 286722878 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector         Size         Last Sector
  0        usr    wm                34      136.72GB          286722911
  1 unassigned    wm                 0           0               0
  2 unassigned    wm                 0           0               0
  3 unassigned    wm                 0           0               0
  4 unassigned    wm                 0           0               0
  5 unassigned    wm                 0           0               0
  6 unassigned    wm                 0           0               0
  8   reserved    wm         286722912        8.00MB          286739295
```

Para utilizar un disco entero, se le debe asignar un nombre de acuerdo con la convención `/dev/dsk/cNtNdN` . Algunos controladores de terceros utilizan otra convención de asignación de nombres o sitúan discos en una ubicación diferente de la del directorio `/dev/dsk`. Para utilizar estos discos, debe etiquetarlos manualmente y proporcionar un segmento a ZFS.

ZFS aplica una etiqueta EFI cuando crea una agrupación de almacenamiento con discos completos. Para obtener más información sobre etiquetas EFI, consulte "EFI Disk Label" de *System Administration Guide: Devices and File Systems*.

Se debe crear un disco destinado a una agrupación raíz ZFS con una etiqueta SMI, no EFI. Puede volver a etiquetar un disco con una etiqueta SMI mediante el uso del comando `format -e`.

Los discos se pueden especificar mediante una ruta completa, como `/dev/dsk/c1t0d0`, o un nombre abreviado compuesto del nombre de dispositivo en el directorio `/dev/dsk`, por ejemplo `c1t0d0`. A continuación puede ver algunos nombres de disco válidos:

- `c1t0d0`
- `/dev/dsk/c1t0d0`
- `/dev/foo/disk`

La forma más sencilla de crear agrupaciones de almacenamiento de ZFS es usar todo el disco físico. Las configuraciones de ZFS se vuelven más complejas de forma progresiva respecto a administración, fiabilidad y rendimiento, cuando se crean agrupaciones de segmentos de discos, LUN (unidades lógicas) en matrices RAID de hardware o volúmenes presentados por administradores de volúmenes basados en software. Las consideraciones siguientes pueden ayudar a determinar la configuración de ZFS con otras soluciones de almacenamiento de hardware o software:

- Si crea una configuración de ZFS sobre unidades LUN a partir de matrices RAID de hardware, debe comprender la relación entre las características de redundancia de ZFS y las de redundancia ofrecidas por la matriz. Determinadas configuraciones pueden dar una redundancia y un rendimiento adecuados, pero otras quizá no lo hagan.
- Puede crear dispositivos lógicos para ZFS mediante volúmenes presentados por administradores de volúmenes basados en software como Solaris Volume Manager (SVM) o Veritas Volume Manager (VxVM). Sin embargo, estas configuraciones no se recomiendan. Aunque ZFS funcione correctamente en estos dispositivos, podría presentar un rendimiento no del todo satisfactorio.

Para obtener información adicional sobre las recomendaciones de agrupaciones de almacenamiento, consulte el sitio sobre métodos recomendados para ZFS:

`http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide`

Los discos se identifican por la ruta e ID de dispositivo, si lo hay. En sistemas donde hay información de ID de dispositivo disponible, este método de identificación permite volver a configurar los dispositivos sin tener que actualizar ZFS. Debido a que los procedimientos de generación y administración de ID de dispositivos pueden variar de un sistema a otro, se recomienda exportar la agrupación antes de mover dispositivos (por ejemplo, trasladar un disco de un controlador a otro). Un evento del sistema como, por ejemplo, una actualización de firmware u otro cambio de hardware, podría cambiar el ID de dispositivo en la agrupación de almacenamiento de ZFS y hacer que los dispositivos no estén disponibles.

Uso de segmentos en una agrupación de almacenamiento de ZFS

Los discos se pueden etiquetar con una etiqueta Solaris VTOC (SMI) tradicional cuando se crea una agrupación de almacenamiento con un segmento de disco.

Para una agrupación raíz ZFS de inicio, los discos de la agrupación deben contener segmentos y deben etiquetarse con una etiqueta SMI. La configuración más sencilla es establecer la capacidad de todo el disco en el segmento 0 y utilizar ese segmento para la agrupación raíz.

En un sistema basado en SPARC, un disco de 72 GB tiene 68 GB de espacio utilizable ubicados en el segmento 0, tal y como se muestra en la siguiente salida de `format`.

```
# format
.
.
.
Specify disk (enter its number): 4
selecting c1t1d0
partition> p
Current partition table (original):
Total disk cylinders available: 14087 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 - 14086       68.35GB    (14087/0/0) 143349312
  1 unassigned    wm       0                0         (0/0/0)             0
  2     backup    wm       0 - 14086       68.35GB    (14087/0/0) 143349312
  3 unassigned    wm       0                0         (0/0/0)             0
  4 unassigned    wm       0                0         (0/0/0)             0
  5 unassigned    wm       0                0         (0/0/0)             0
  6 unassigned    wm       0                0         (0/0/0)             0
  7 unassigned    wm       0                0         (0/0/0)             0
```

En un sistema basado en x86, un disco de 72 GB tiene 68 GB de espacio utilizable ubicados en el segmento 0, tal y como se muestra en la siguiente salida de `format`. En el segmento 8 se incluye una pequeña cantidad de información de inicio. El segmento 8 no requiere administración y no se puede cambiar.

```
# format
.
.
.
selecting c1t0d0
partition> p
Current partition table (original):
Total disk cylinders available: 49779 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       1 - 49778       68.36GB    (49778/0/0) 143360640
  1 unassigned    wu       0                0         (0/0/0)             0
  2     backup    wm       0 - 49778       68.36GB    (49779/0/0) 143363520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
  8       boot    wu       0 -     0        1.41MB    (1/0/0)          2880
  9 unassigned    wu       0                0         (0/0/0)             0
```

Uso de archivos en una agrupación de almacenamiento de ZFS

ZFS también permite utilizar los archivos UFS como dispositivos virtuales en la agrupación de almacenamiento. Esta función se aplica sobre todo a verificaciones y pruebas sencillas, no es apta la producción. El motivo es que **cualquier uso de los archivos se basa en el sistema de**

archivos subyacente por motivos de coherencia. Si crea una agrupación ZFS respaldada por archivos en un sistema de archivos UFS, de forma implícita depende de UFS para garantizar la corrección y una semántica síncrona.

Sin embargo, los archivos pueden ser bastante útiles al probar ZFS por primera vez o experimentar con configuraciones más complejas cuando no hay suficientes dispositivos físicos. Se deben especificar todos los archivos como rutas completas y deben tener al menos 64 MB de tamaño.

Funciones de repetición de una agrupación de almacenamiento de ZFS

ZFS proporciona redundancia de datos y propiedades de autocorrección en configuraciones reflejadas y RAID-Z.

Configuración reflejada de agrupaciones de almacenamiento

Una configuración reflejada de agrupación de almacenamiento necesita al menos dos discos, preferiblemente en controladores independientes. En una configuración reflejada se pueden utilizar muchos discos. Asimismo, puede crear más de un reflejo en cada agrupación. Conceptualmente hablando, una configuración reflejada sencilla tendría un aspecto similar al siguiente:

```
mirror c1t0d0 c2t0d0
```

Desde un punto de vista conceptual, una configuración reflejada más compleja tendría un aspecto similar al siguiente:

```
mirror c1t0d0 c2t0d0 c3t0d0 mirror c4t0d0 c5t0d0 c6t0d0
```

Para obtener información sobre cómo crear agrupaciones de almacenamiento reflejadas, consulte "Creación de una agrupación de almacenamiento reflejado" en la página 73.

Configuración de agrupaciones de almacenamiento RAID-Z

Además de una configuración reflejada de agrupación de almacenamiento, ZFS ofrece una configuración de RAID-Z con tolerancia a fallos de paridad sencilla, doble o triple. RAID-Z de paridad sencilla (`raidz` o `raidz1`) es similar a RAID-5. RAID-Z de paridad doble (`raidz2`) es similar a RAID-6.

Para obtener más información sobre RAIDZ-3 (`raidz3`), consulte el blog siguiente:

`http://blogs.sun.com/ahl/entry/triple_parity_raid_z`

Todos los algoritmos tradicionales similares a RAID-5 (RAID-4, RAID-6, RDP y par-impar, por ejemplo) tienen un problema conocido como "error de escritura por caída del sistema de RAID-5". Si sólo se escribe parte de una distribución de discos en bandas de RAID-5 y la alimentación se interrumpe antes de que todos los bloques se hayan escrito en el disco, la paridad permanece sin sincronizarse con los datos, y por eso deja de ser útil (a menos que se sobrescriba con una escritura posterior de todas las bandas). En RAID-Z, ZFS utiliza repartos de discos en bandas de RAID de ancho variable, de manera que todas las escrituras son de reparto total de discos en bandas. Este diseño sólo es posible porque ZFS integra el sistema de archivos y la administración de dispositivos de manera que los metadatos del sistema de archivos tengan suficiente información sobre el modelo de redundancia de los datos subyacentes para controlar los repartos de discos en bandas de RAID de anchura variable. RAID-Z es la primera solución exclusiva de software en el mundo para el error de escritura por caída del sistema de RAID-5.

Una configuración de RAID-Z con N discos de tamaño X con discos de paridad P puede contener aproximadamente (N-P)*X bytes, así como admitir uno o más dispositivos P con errores antes de que se comprometa la integridad de los datos. Para la configuración de RAID-Z de paridad sencilla se necesita un mínimo de dos discos y al menos tres para la configuración de RAID-Z de paridad doble. Por ejemplo, si tiene tres discos en una configuración de RAID-Z de paridad sencilla, los datos de la paridad ocupan un espacio equivalente a uno de los tres discos. Para crear una configuración de RAID-Z no se necesita hardware especial.

Conceptualmente hablando, una configuración de RAID-Z con tres discos tendría un aspecto similar al siguiente:

```
raidz c1t0d0 c2t0d0 c3t0d0
```

Mientras que una configuración reflejada más compleja tendría un aspecto similar al siguiente:

```
raidz c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0 c6t0d0 c7t0d0 raidz c8t0d0 c9t0d0 c10t0d0 c11t0d0
c12t0d0 c13t0d0 c14t0d0
```

Si desea crear una configuración de RAID-Z con muchos discos, puede ser conveniente dividir los discos en varios grupos. Por ejemplo, una configuración de RAID-Z con 14 discos se puede

dividir en dos grupos de 7 discos. En principio, las configuraciones de RAID-Z con agrupaciones de un solo dígito de discos funcionan mejor.

Para obtener información sobre cómo crear una agrupación de almacenamiento de RAID-Z, consulte "Creación de una agrupación de almacenamiento de RAID-Z" en la página 74.

Para obtener más información sobre cómo elegir entre una configuración reflejada o una de RAID-Z en función del espacio y el rendimiento, consulte el blog siguiente:

`http://blogs.sun.com/roch/entry/when_to_and_not_to`

Para obtener información adicional sobre las recomendaciones de agrupaciones de almacenamiento de RAID-Z, consulte el sitio sobre métodos recomendados para ZFS:

`http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide`

Agrupación de almacenamiento híbrido de ZFS

La agrupación de almacenamiento híbrido de ZFS, disponible en la serie de productos de Oracle Sun Storage 7000, es una agrupación de almacenamiento especial que combina DRAM, SSD y HDD con el fin de mejorar el rendimiento y aumentar la capacidad, al tiempo que se reduce el consumo de energía. Con la interfaz de administración de este producto, puede seleccionar la configuración de redundancia de ZFS de la agrupación de almacenamiento y administrar fácilmente otras opciones de configuración.

Para obtener más información acerca de este producto, consulte la *Sun Storage Unified Storage System Administration Guide*.

Datos de recuperación automática en una configuración redundante

ZFS ofrece soluciones para datos de recuperación automática en una configuración de RAID-Z o reflejada.

Si se detecta un bloque de datos incorrectos, ZFS no sólo recupera los datos correctos de otra copia redundante, sino que además repara los datos incorrectos sustituyéndolos por la copia correcta

Reparto dinámico de discos en bandas en una agrupación de almacenamiento

ZFS reparte dinámicamente los datos de los discos en bandas entre todos los dispositivos virtuales de nivel superior. La elección de ubicación de los datos se efectúa en el momento de la escritura, por lo que en el momento de la asignación no se crean bandas de ancho fijo.

Cuando se agregan a una agrupación dispositivos virtuales nuevos, ZFS asigna datos gradualmente al nuevo dispositivo con el fin de mantener el rendimiento y las normas de asignación de espacio. Cada dispositivo virtual puede ser también un reflejo o un dispositivo de RAID-Z que contenga otros archivos o dispositivos de discos. Esta configuración ofrece flexibilidad a la hora de controlar las características predeterminadas de la agrupación. Por ejemplo, puede crear las configuraciones siguientes a partir de cuatro discos:

- Cuatro discos que utilicen reparto dinámico de discos en bandas
- Una configuración de RAID-Z de cuatro vías
- Dos reflejos de dos vías que utilicen reparto dinámico de discos en bandas

Aunque ZFS admite la combinación de diversos tipos de dispositivos virtuales en la misma agrupación, debe evitar hacerlo. Por ejemplo, puede crear una agrupación con un reflejo de dos vías y una configuración de RAID-Z de tres vías. Sin embargo, la tolerancia a errores es como el peor de los dispositivos virtuales de que disponga, en este caso RAID-Z. La práctica recomendada es utilizar dispositivos virtuales de nivel superior del mismo tipo con idéntico nivel de redundancia en cada dispositivo.

Creación y destrucción de agrupaciones de almacenamiento de ZFS

Las secciones siguientes describen distintas situaciones de creación y destrucción de agrupaciones de almacenamiento de ZFS:

- "Creación de una agrupación de almacenamiento de ZFS" en la página 72
- "Visualización de información de dispositivos virtuales de agrupaciones de almacenamiento" en la página 77
- "Administración de errores de creación de agrupaciones de almacenamiento de ZFS" en la página 79
- "Destrucción de agrupaciones de almacenamiento de ZFS" en la página 82

La creación y la destrucción de agrupaciones son procesos fáciles y rápidos. Sin embargo, estas operaciones se deben efectuar con cuidado. Aunque las comprobaciones se efectúan para impedir el uso de dispositivos se están usando en una nueva agrupación, ZFS no puede saber siempre si un dispositivo ya se está utilizando. La destrucción de una agrupación es más fácil que crear uno. Utilice `zpool destroy` con precaución. Este comando sencillo tiene importantes consecuencias.

Creación de una agrupación de almacenamiento de ZFS

Para crear una agrupación de almacenamiento, utilice el comando `zpool create`. Este comando toma un nombre de agrupación y cualquier cantidad de dispositivos virtuales como

argumentos. El nombre de la agrupación debe atenerse a los requisitos de denominación indicados en "Requisitos de asignación de nombres de componentes de ZFS" en la página 51.

Creación de una agrupación de almacenamiento básico

El comando siguiente crea un recurso con el nombre `tank` que se compone de los discos `c1t0d0` y `c1t1d0`:

```
# zpool create tank c1t0d0 c1t1d0
```

Los nombres de dispositivo que representan los discos completos se encuentran en el directorio `/dev/dsk`; ZFS los etiqueta correspondientemente para que contengan un segmento único y de gran tamaño. Los datos se reparten dinámicamente en ambos discos.

Creación de una agrupación de almacenamiento reflejado

Para crear una agrupación reflejada, utilice la palabra clave `mirror`, seguida de varios dispositivos de almacenamiento que incluirán el reflejo. Se pueden especificar varios reflejos si se repite la palabra clave `mirror` en la línea de comandos. El comando siguiente crea una agrupación con dos reflejos de dos vías:

```
# zpool create tank mirror c1d0 c2d0 mirror c3d0 c4d0
```

La segunda palabra clave `mirror` indica que se especifica un nuevo dispositivo virtual de nivel superior. Los datos se colocan dinámicamente en bandas en los dos reflejos, con la correspondiente redundancia de datos en cada disco.

Para obtener más información sobre configuraciones reflejadas recomendadas, visite la página web siguiente:

`http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide`

En la actualidad, en una configuración reflejada de ZFS son posibles las operaciones siguientes:

- Agregar otro conjunto de discos de nivel superior adicional (vdev) a una configuración reflejada existente. Para obtener más información, consulte "Adición de dispositivos a una agrupación de almacenamiento" en la página 83.
- Conectar discos adicionales a una configuración reflejada. Conectar discos adicionales a una configuración no repetida para crear una configuración reflejada. Para obtener más información, consulte "Conexión y desconexión de dispositivos en una agrupación de almacenamiento" en la página 88.
- Reemplazar uno o varios discos de una configuración reflejada existente si los discos de sustitución son mayores o iguales que el dispositivo que se va a reemplazar. Para obtener más información, consulte "Sustitución de dispositivos en una agrupación de almacenamiento" en la página 96.

- Desconectar un disco de una configuración reflejada si los demás dispositivos proporcionan a la configuración la redundancia necesaria. Para obtener más información, consulte "Conexión y desconexión de dispositivos en una agrupación de almacenamiento" en la página 88.
- División de una configuración reflejada mediante la desconexión de uno de los discos para crear una agrupación nueva idéntica. Para obtener más información, consulte "Creación de una nueva agrupación mediante la división de una agrupación de almacenamiento de ZFS reflejada" en la página 90.

No puede quitarse de una agrupación de almacenamiento reflejada un dispositivo que no sea de registro o de caché. Para esta función se presenta un RFE.

Creación de una agrupación raíz ZFS

Puede instalar e iniciar el sistema a partir de ZFS desde un sistema de archivos raíz ZFS. Revise la siguiente información de configuración de agrupaciones raíz:

- Los discos utilizados para la agrupación raíz deben tener una etiqueta VTOC (SMI) y la agrupación se debe crear con segmentos de discos.
- La agrupación raíz debe crearse como configuración reflejada o una configuración de un solo disco. No se pueden agregar discos adicionales para crear varios dispositivos virtuales reflejados de nivel superior mediante el comando `zpool add`, pero se puede ampliar un dispositivo virtual reflejado mediante el comando `zpool attach`.
- No se admite una configuración RAID-Z o repartida.
- Una agrupación raíz no puede tener un dispositivo de registro independiente.
- Si intenta utilizar una configuración no admitida para una agrupación raíz, verá mensajes similares a éstos:

```
ERROR: ZFS pool <pool-name> does not support boot environments
```

```
# zpool add -f rpool log c0t6d0s0
cannot add to 'rpool': root pool can not have multiple vdevs or separate logs
```

Para más información sobre cómo instalar e iniciar un sistema de archivos raíz ZFS, consulte el Capítulo 5, "Instalación e inicio de un sistema de archivos raíz ZFS Oracle Solaris".

Creación de una agrupación de almacenamiento de RAID-Z

Una agrupación de RAID-Z de paridad sencilla se crea del mismo modo que una agrupación reflejada, excepto que se utiliza la palabra clave `raidz` o `raidz1` en lugar de `mirror`. El ejemplo siguiente muestra cómo crear una agrupación con un único dispositivo de RAID-Z que se compone de cinco discos:

```
# zpool create tank raidz c1t0d0 c2t0d0 c3t0d0 c4t0d0 /dev/dsk/c5t0d0
```

Este ejemplo muestra que los discos se pueden especificar con sus nombres de dispositivo abreviados o completos. `/dev/dsk/c5t0d0` y `c5t0d0` hacen referencia al mismo disco.

Puede crear una configuración de RAID-Z de paridad doble mediante la palabra clave `raidz2` o `raidz3` al crear la agrupación. Por ejemplo:

```
# zpool create tank raidz2 c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0
# zpool status -v tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          raidz2-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0
            c3t0d0  ONLINE       0     0     0
            c4t0d0  ONLINE       0     0     0
            c5t0d0  ONLINE       0     0     0

errors: No known data errors

# zpool create tank raidz3 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0 c6t0d0 c7t0d0
# zpool status -v tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          raidz3-0  ONLINE       0     0     0
            c0t0d0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0
            c3t0d0  ONLINE       0     0     0
            c4t0d0  ONLINE       0     0     0
            c5t0d0  ONLINE       0     0     0
            c6t0d0  ONLINE       0     0     0
            c7t0d0  ONLINE       0     0     0

errors: No known data errors
```

En la actualidad, en una configuración RAID-Z de ZFS son posibles las operaciones siguientes:

- Agregar a una configuración RAID-Z existente otro conjunto de discos para un dispositivo virtual de nivel superior. Para obtener más información, consulte "Adición de dispositivos a una agrupación de almacenamiento" en la página 83.
- Reemplazar uno o varios discos de una configuración RAID-Z existente si los discos de sustitución son mayores o iguales que el dispositivo que se va a reemplazar. Para obtener más información, consulte "Sustitución de dispositivos en una agrupación de almacenamiento" en la página 96.

Actualmente *no* se permiten las siguientes operaciones en una configuración RAID-Z:

- Conectar un disco adicional a una configuración de RAID-Z.

- Suprimir un disco de una configuración de RAID-Z, excepto cuando se trate de un disco que se sustituye por un disco de reserva.
- No puede quitarse de una configuración de RAID-Z un dispositivo que no sea de registro o caché. Para esta función se presenta un RFE.

Para obtener más información sobre una configuración de RAID-Z, consulte "Configuración de agrupaciones de almacenamiento RAID-Z" en la página 70.

Creación de una agrupación de almacenamiento de ZFS con dispositivos de registro

De forma predeterminada, ZIL se asigna a partir de bloques de la agrupación principal. Sin embargo, el rendimiento puede mejorar si se usan dispositivos de registro independientes, por ejemplo NVRAM o un disco dedicado. Para obtener más información sobre dispositivos de registro ZFS, consulte "Configuración de dispositivos de registro de ZFS independientes" en la página 34.

Puede crear un dispositivo de registro ZFS durante la creación de la agrupación o una vez creada.

El ejemplo siguiente muestra cómo crear una agrupación de almacenamiento reflejada con dispositivos de registro reflejados:

```
# zpool create datap mirror c1t1d0 c1t2d0 mirror c1t3d0 c1t4d0 log mirror c1t5d0 c1t8d0
# zpool status datap
  pool: datap
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        datap       ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0
            c1t4d0  ONLINE       0     0     0
        logs
          mirror-2  ONLINE       0     0     0
            c1t5d0  ONLINE       0     0     0
            c1t8d0  ONLINE       0     0     0

errors: No known data errors
```

Para obtener información sobre la recuperación de un error en un dispositivo de registro, consulte el Ejemplo 11–2.

Creación de una agrupación de almacenamiento de ZFS con dispositivos caché

Puede crear una agrupación de almacenamiento con dispositivos caché para guardar en éstos datos de la agrupación de almacenamiento. Por ejemplo:

```
# zpool create tank mirror c2t0d0 c2t1d0 c2t3d0 cache c2t5d0 c2t8d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0
        cache
          c2t5d0    ONLINE       0     0     0
          c2t8d0    ONLINE       0     0     0

errors: No known data errors
```

Tenga en cuenta los siguientes puntos antes de decidir si se debe crear una agrupación de almacenamiento de ZFS con dispositivos caché:

- El uso de dispositivos caché optimiza el rendimiento con cargas de trabajo de lectura aleatorias de contenido principalmente estático.
- La capacidad y las lecturas se pueden supervisar mediante el comando `zpool iostat`.
- Se pueden añadir varios dispositivos caché cuando se crea la agrupación. Asimismo se pueden añadir y eliminar después de crearse la agrupación. Para obtener más información, consulte el Ejemplo 4–4.
- Los dispositivos caché no se pueden reflejar ni pueden formar parte de una configuración de RAID-Z.
- Si se encuentra un error de lectura en un dispositivo caché, la E/S de lectura se vuelve a enviar al dispositivo de agrupación de almacenamiento original, que puede formar parte de una configuración de RAID-Z o reflejada. El contenido de los dispositivos caché se considera volátil, de forma similar a otras memorias caché del sistema.

Visualización de información de dispositivos virtuales de agrupaciones de almacenamiento

Cada agrupación de almacenamiento contiene uno o más dispositivos virtuales. Un *dispositivo virtual* es una representación interna de la agrupación de almacenamiento que describe la disposición del almacenamiento físico y sus características predeterminadas. Un dispositivo

virtual representa los archivos o dispositivos de disco que se utilizan para crear la agrupación de almacenamiento. Una agrupación puede tener en la parte superior de la configuración cualquier cantidad de dispositivos virtuales, denominados *vdev de nivel superior*.

Si el dispositivo virtual de nivel superior contiene dos o más dispositivos físicos, la configuración ofrece redundancia de datos como reflejo o dispositivo virtual RAID-Z. Estos dispositivos virtuales se componen de discos, segmentos de discos o archivos. Un repuesto es un caso especial de dispositivo virtual que hace un seguimiento de repuestos disponibles para una agrupación.

El ejemplo siguiente muestra cómo crear una agrupación formada por dos dispositivos virtuales de nivel superior, cada uno de los cuales es un reflejos de dos discos:

```
# zpool create tank mirror c1d0 c2d0 mirror c3d0 c4d0
```

El ejemplo siguiente muestra cómo crear una agrupación formada por un dispositivo virtual de nivel superior de cuatro discos:

```
# zpool create mypool raidz2 c1d0 c2d0 c3d0 c4d0
```

Se puede agregar otro dispositivo virtual de nivel superior a esta agrupación mediante el comando `zpool add`. Por ejemplo:

```
# zpool add mypool raidz2 c2d1 c3d1 c4d1 c5d1
```

Los discos, segmentos de discos o archivos que se utilizan en agrupaciones no redundantes funcionan como dispositivos virtuales de nivel superior. Las agrupaciones de almacenamiento suelen contener diversos dispositivos virtuales de nivel superior. ZFS reparte dinámicamente los discos en bandas entre todos los dispositivos virtuales de nivel superior en una agrupación.

Los dispositivos virtuales y físicos que se incluyen en una agrupación de almacenamiento de ZFS se muestran con el comando `zpool status`. Por ejemplo:

```
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            c0t2d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
            c0t3d0  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0

errors: No known data errors
```

Administración de errores de creación de agrupaciones de almacenamiento de ZFS

Los errores de creación de agrupaciones pueden deberse a diversos motivos. Algunos de ellos son obvios (por ejemplo, un dispositivo especificado que no existe), mientras que otros no lo son tanto.

Detección de dispositivos en uso

Antes de dar formato a un dispositivo, ZFS determina si el disco lo está utilizando ZFS o cualquier otro componente del sistema operativo. Si el disco está en uso, puede haber errores como el siguiente:

```
# zpool create tank c1t0d0 c1t1d0
invalid vdev specification
use '-f' to override the following errors:
/dev/dsk/c1t0d0s0 is currently mounted on /. Please see umount(1M).
/dev/dsk/c1t0d0s1 is currently mounted on swap. Please see swap(1M).
/dev/dsk/c1t1d0s0 is part of active ZFS pool zeepool. Please see zpool(1M).
```

Algunos errores pueden omitirse mediante la opción `-f`, pero no es algo aplicable a la mayoría. Las condiciones siguientes no pueden omitirse mediante la opción `-f`; se deben corregir manualmente:

Sistema de archivos montado — El disco o uno de sus segmentos contiene un sistema de archivos que está montado. Para corregir este error, utilice el comando `umount`.

Sistema de archivos en /etc/vfstab — El disco contiene un sistema de archivos que se muestra en el archivo `/etc/vfstab`, pero el sistema de archivos no está montado. Para corregir este error, suprima la línea del archivo `/etc/vfstab` o conviértala en comentario.

Dispositivo de volcado dedicado — El disco se utiliza como dispositivo de volcado dedicado para el sistema. Para corregir este error, utilice el comando `dumpadm`.

Parte de una agrupación de ZFS — El disco o archivo es parte de una agrupación de almacenamiento de ZFS activa. Para corregir este error, utilice el comando `zpool destroy` para destruir la otra agrupación, si ya no se necesita. También puede utilizar el comando `zpool detach` para desvincular el disco de la otra agrupación. Sólo se puede desvincular un disco de una agrupación de almacenamiento reflejada.

Las siguientes comprobaciones en uso son advertencias útiles; se pueden anular mediante la opción `-f` para crear la agrupación:

Contiene un sistema de archivos	El disco contiene un sistema de archivos conocido, aunque no está montado y no parece que se utilice.
Parte de volumen	El disco es parte de un volumen de Solaris Volume Manager.
Actualización automática	El disco se utiliza como entorno de inicio alternativo para Actualización automática de Oracle Solaris.
Parte de agrupación de ZFS exportado	El disco es parte de una agrupación de almacenamiento que se ha exportado o suprimido manualmente de un sistema. En el último caso, se informa de que la agrupación es `potencialmente activo`, ya que el disco quizá sea o no una unidad conectada a la red que otro sistema utiliza. Actúe con precaución al anular una agrupación potencialmente activa.

El ejemplo siguiente muestra la forma de utilizar la opción `-f`:

```
# zpool create tank c1t0d0
invalid vdev specification
use '-f' to override the following errors:
/dev/dsk/c1t0d0s0 contains a ufs filesystem.
# zpool create -f tank c1t0d0
```

En lugar de utilizar la opción `-f` es preferible corregir los errores.

Niveles de repetición no coincidentes

No se recomienda crear agrupaciones con dispositivos virtuales de niveles de repetición diferentes. El comando `zpool` impide la creación involuntaria de una agrupación con niveles de redundancia que no coinciden. Si intenta crear una agrupación con una configuración de ese tipo, aparecen errores similares al siguiente:

```
# zpool create tank c1t0d0 mirror c2t0d0 c3t0d0
invalid vdev specification
use '-f' to override the following errors:
mismatched replication level: both disk and mirror vdevs are present
# zpool create tank mirror c1t0d0 c2t0d0 mirror c3t0d0 c4t0d0 c5t0d0
invalid vdev specification
use '-f' to override the following errors:
mismatched replication level: 2-way mirror and 3-way mirror vdevs are present
```

Puede anular estos errores con la opción `-f`, pero debería evitar esta práctica. El comando también advierte sobre la creación de una agrupación de RAID-Z o reflejada mediante dispositivos de diversos tamaños. Aunque esta configuración se permite, los niveles sin

correspondencia de redundancia generan espacio sin usar en disco en el dispositivo de mayor tamaño. Se necesita la opción `-f` para anular la advertencia.

Ensayo de creación de una agrupación de almacenamiento

Los intentos de creación de agrupación pueden fallar de modo imprevisto y formas diferentes; la aplicación de formato a discos es una acción potencialmente perjudicial. Por ello, el comando `zpool create` tiene la opción adicional `-n` que simula la creación de la agrupación sin escribir realmente en el dispositivo. Esta opción de *ensayo* realiza la comprobación del dispositivo en uso y la validación de nivel de repetición, y notifica si se producen errores en el proceso. Si no se encuentran errores, se genera una salida similar a la siguiente:

```
# zpool create -n tank mirror c1t0d0 c1t1d0
would create 'tank' with the following layout:

        tank
          mirror
            c1t0d0
            c1t1d0
```

Algunos errores no se pueden detectar sin crear la agrupación. El caso más habitual es la especificación del mismo dispositivo dos veces en la misma configuración. Este error puede pasar desapercibido si no se escriben los datos, por lo que el comando `zpool create -n` podría notificar que la operación es correcta y aun así no conseguir crear la agrupación cuando el comando se ejecuta sin esta opción.

Punto de montaje predeterminado para agrupaciones de almacenamiento

Cuando se crea una agrupación, el punto de montaje predeterminado del conjunto de datos de nivel superior es */nombre_agrupación*. Este directorio no debe existir o debe estar vacío. Si el directorio no existe, se crea automáticamente. Si está vacío, el conjunto de datos raíz se monta en la parte superior del directorio ya creado. Para crear una agrupación con un punto de montaje predeterminado diferente, utilice la opción `-m` del comando `zpool create`. Por ejemplo:

```
# zpool create home c1t0d0
default mountpoint '/home' exists and is not empty
use '-m' option to provide a different default
# zpool create -m /export/zfs home c1t0d0
```

Este comando crea la agrupación `home` y el conjunto de datos `home` con un punto de montaje de `/export/zfs`.

Para obtener más información sobre los puntos de montaje, consulte "Administración de puntos de montaje de ZFS" en la página 211.

Destrucción de agrupaciones de almacenamiento de ZFS

Para destruir agrupaciones se utiliza el comando `zpool destroy`. Este comando destruye la agrupación aunque contenga conjuntos de datos montados.

```
# zpool destroy tank
```

Precaución – Ponga el máximo cuidado al destruir una agrupación. Asegúrese de que se va a destruir la agrupación correcta y guarde siempre copias de los datos. Si destruye involuntariamente la agrupación incorrecta, puede intentar recuperarla. Para obtener más información, consulte "Recuperación de agrupaciones de almacenamiento de ZFS destruidas" en la página 121.

Destrucción de una agrupación con dispositivos con errores

La destrucción de una agrupación requiere que los datos se escriban en el disco para indicar que la agrupación ya no es válida. Esta información del estado impide que los dispositivos aparezcan como una agrupación potencial cuando efectúa una importación. Si uno o más dispositivos dejan de estar disponibles, sigue siendo posible destruir la agrupación. Pero la información de estado necesaria no se escribirá en estos dispositivos no disponibles.

Cuando se reparan adecuadamente, estos dispositivos se notifican como *potencialmente activos* cuando se crea una agrupación. Se incluyen como dispositivos válidos si se buscan agrupaciones para importar. Si una agrupación tiene tantos dispositivos con errores que la propia agrupación aparece con errores (lo que significa que el dispositivo virtual de nivel superior es incorrecto), el comando imprime una advertencia y no se puede completar sin la opción `-f`. Esta opción es necesaria porque la agrupación no se puede abrir, de manera que no se sabe si los datos están o no almacenados allí. Por ejemplo:

```
# zpool destroy tank
cannot destroy 'tank': pool is faulted
use '-f' to force destruction anyway
# zpool destroy -f tank
```

Para obtener más información sobre la situación de dispositivos y agrupaciones, consulte "Cómo determinar el estado de las agrupaciones de almacenamiento de ZFS" en la página 113.

Para obtener más información sobre importación de agrupaciones, consulte "Importación de agrupaciones de almacenamiento de ZFS" en la página 120.

Administración de dispositivos en agrupaciones de almacenamiento de ZFS

La mayor parte de la información básica relacionada con los dispositivos se puede consultar en "Componentes de una agrupación de almacenamiento de ZFS" en la página 65. Después de crear una agrupación, puede efectuar diversas tareas para administrar los dispositivos físicos en ella.

Adición de dispositivos a una agrupación de almacenamiento

Puede agregar espacio en el disco a una agrupación de forma dinámica, incorporando un nuevo dispositivo virtual de nivel superior. Este espacio está inmediatamente disponible para todos los conjuntos de datos de la agrupación. Para agregar un dispositivo virtual a una agrupación, utilice el comando `zpool add`. Por ejemplo:

```
# zpool add zeepool mirror c2t1d0 c2t2d0
```

El formato para especificar dispositivos virtuales es el mismo que para el comando `zpool create`. Los dispositivos se comprueban para determinar si se utilizan y el comando no puede cambiar el nivel de redundancia sin la opción `-f`. El comando también es compatible con la opción `-n` de manera que puede ejecutar un ensayo. Por ejemplo:

```
# zpool add -n zeepool mirror c3t1d0 c3t2d0
would update 'zeepool' to the following configuration:
      zeepool
        mirror
            c1t0d0
            c1t1d0
        mirror
            c2t1d0
            c2t2d0
        mirror
```

```
            c3t1d0
            c3t2d0
```

La sintaxis de este comando agregaría dispositivos reflejados `c3t1d0` y `c3t2d0` a la configuración existente de la agrupación `zeepool`.

Para obtener más información sobre cómo validar dispositivos virtuales, consulte "Detección de dispositivos en uso" en la página 79.

EJEMPLO 4–1 Adición de discos a una configuración de ZFS reflejada

En el ejemplo siguiente se agrega un reflejo a otro reflejo de ZFS ya existente en un sistema Sun Fire x4500 de Oracle.

```
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            c0t2d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0

errors: No known data errors
# zpool add tank mirror c0t3d0 c1t3d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            c0t2d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
            c0t3d0  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0

errors: No known data errors
```

EJEMPLO 4–2 Adición de discos a una configuración de RAID-Z

Se pueden agregar discos adicionales de modo similar a una configuración de RAID-Z. El ejemplo siguiente muestra cómo convertir una agrupación de almacenamiento con un dispositivo RAID–Z que contiene tres discos en una agrupación de almacenamiento con dos dispositivos RAID-Z con tres discos cada uno.

```
# zpool status rzpool
  pool: rzpool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        rzpool      ONLINE       0     0     0
          raidz1-0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0
            c1t4d0  ONLINE       0     0     0

errors: No known data errors
# zpool add rzpool raidz c2t2d0 c2t3d0 c2t4d0
# zpool status rzpool
  pool: rzpool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        rzpool      ONLINE       0     0     0
          raidz1-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0
          raidz1-1  ONLINE       0     0     0
            c2t2d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0
            c2t4d0  ONLINE       0     0     0

errors: No known data errors
```

EJEMPLO 4–3 Adición y eliminación de un dispositivo de registro reflejado

En el ejemplo siguiente se muestra cómo agregar un dispositivo de registro reflejado a una agrupación de almacenamiento reflejada.Para obtener más información sobre cómo utilizar dispositivos de registro en la agrupación de almacenamiento, consulte "Configuración de dispositivos de registro de ZFS independientes" en la página 34.

```
# zpool status newpool
  pool: newpool
 state: ONLINE
 scrub: none requested
config:
```

EJEMPLO 4–3 Adición y eliminación de un dispositivo de registro reflejado *(Continuación)*

```
        NAME        STATE     READ WRITE CKSUM
        newpool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
            c0t5d0  ONLINE       0     0     0

errors: No known data errors
# zpool add newpool log mirror c0t6d0 c0t7d0
# zpool status newpool
  pool: newpool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        newpool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
            c0t5d0  ONLINE       0     0     0
        logs
          mirror-1  ONLINE       0     0     0
            c0t6d0  ONLINE       0     0     0
            c0t7d0  ONLINE       0     0     0

errors: No known data errors
```

Puede vincular un dispositivo de registro a uno ya creado para crear un dispositivo de registro reflejado. Esta operación es idéntica a la de vincular un dispositivo en una agrupación de almacenamiento sin reflejar.

Los dispositivos de registro se pueden eliminar utilizando el comando `zpool remove`. El dispositivo de registro reflejado en el ejemplo anterior se puede eliminar mediante la especificación del argumento `mirror-1`. Por ejemplo:

```
# zpool remove newpool mirror-1
# zpool status newpool
  pool: newpool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        newpool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
            c0t5d0  ONLINE       0     0     0

errors: No known data errors
```

Si su configuración de agrupación sólo contiene un dispositivo de registro, para eliminar éste tendrá que especificar el nombre del dispositivo. Por ejemplo:

EJEMPLO 4–3 Adición y eliminación de un dispositivo de registro reflejado *(Continuación)*

```
# zpool status pool
  pool: pool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        pool        ONLINE       0     0     0
          raidz1-0  ONLINE       0     0     0
            c0t8d0  ONLINE       0     0     0
            c0t9d0  ONLINE       0     0     0
        logs
          c0t10d0   ONLINE       0     0     0

errors: No known data errors
# zpool remove pool c0t10d0
```

EJEMPLO 4–4 Cómo añadir y eliminar dispositivos caché

Se pueden agregar a la agrupación de almacenamiento de ZFS y quitarlos si dejan de ser necesarios.

Utilice el comando `zpool add` para agregar dispositivos caché. Por ejemplo:

```
# zpool add tank cache c2t5d0 c2t8d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0
        cache
          c2t5d0    ONLINE       0     0     0
          c2t8d0    ONLINE       0     0     0

errors: No known data errors
```

Los dispositivos caché no se pueden reflejar ni pueden formar parte de una configuración de RAID-Z.

Utilice el comando `zpool remove` para eliminar dispositivos caché. Por ejemplo:

```
# zpool remove tank c2t5d0 c2t8d0
# zpool status tank
 pool: tank
 state: ONLINE
```

EJEMPLO 4–4 Cómo añadir y eliminar dispositivos caché *(Continuación)*

```
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0

errors: No known data errors
```

Actualmente, el comando `zpool remove` sólo admite la eliminación de dispositivos caché, dispositivos de registro y repuestos en marcha. Los dispositivos que forman parte de la configuración de la agrupación reflejada principal se pueden eliminar mediante el comando `zpool detach`. Los dispositivos no redundantes y de RAID-Z no se pueden eliminar de una agrupación.

Para obtener más información sobre cómo utilizar dispositivos caché en una agrupación de almacenamiento de ZFS, consulte "Creación de una agrupación de almacenamiento de ZFS con dispositivos caché" en la página 77.

Conexión y desconexión de dispositivos en una agrupación de almacenamiento

Además del comando `zpool add`, puede utilizar el comando `zpool attach` para agregar un nuevo dispositivo a un dispositivo reflejado o no reflejado existente.

Si va a vincular un disco para crear una agrupación raíz reflejada, consulte "Cómo crear una agrupación raíz reflejada (posterior a la instalación)" en la página 136.

Si va a reemplazar un disco en la agrupación de almacenamiento de ZFS, consulte "Cómo sustituir un disco en la agrupación raíz ZFS" en la página 178.

EJEMPLO 4–5 Conversión de una agrupación de almacenamiento reflejada de dos vías a una reflejada de tres vías

En este ejemplo, `zeepool` es un reflejo de dos vías que se transforma en uno de tres vías mediante la conexión del nuevo dispositivo `c2t1d0` a `c1t1d0`, el que ya existía.

```
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: none requested
config:
```

EJEMPLO 4–5 Conversión de una agrupación de almacenamiento reflejada de dos vías a una reflejada de tres vías *(Continuación)*

```
        NAME        STATE     READ WRITE CKSUM
        zeepool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0

errors: No known data errors
# zpool attach zeepool c1t1d0 c2t1d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Fri Jan  8 12:59:20 2010
config:

        NAME        STATE     READ WRITE CKSUM
        zeepool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0  592K resilvered

errors: No known data errors
```

Si el dispositivo existente forma parte de un reflejo de tres vías, al conectar el nuevo dispositivo se crea un reflejo de cuatro vías, y así sucesivamente. En cualquier caso, el nuevo dispositivo comienza inmediatamente la actualización de la duplicación.

EJEMPLO 4–6 Conversión de una agrupación de almacenamiento de ZFS no redundante a una de ZFS reflejada

También se puede convertir una agrupación de almacenamiento no redundante en una redundante mediante el comando `zpool attach`. Por ejemplo:

```
# zpool create tank c0t1d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:
        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          c0t1d0    ONLINE       0     0     0

errors: No known data errors
# zpool attach tank c0t1d0 c1t1d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Fri Jan  8 14:28:23 2010
config:

        NAME        STATE     READ WRITE CKSUM
```

EJEMPLO 4–6 Conversión de una agrupación de almacenamiento de ZFS no redundante a una de ZFS reflejada *(Continuación)*

```
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0  73.5K resilvered

errors: No known data errors
```

Puede utilizar el comando `zpool detach` para desconectar un dispositivo de una agrupación de almacenamiento reflejada. Por ejemplo:

```
# zpool detach zeepool c2t1d0
```

Pero esta operación fallará si no hay ninguna otra réplica válida de los datos. Por ejemplo:

```
# zpool detach newpool c1t2d0
cannot detach c1t2d0: only applicable to mirror and replacing vdevs
```

Creación de una nueva agrupación mediante la división de una agrupación de almacenamiento de ZFS reflejada

Una agrupación de almacenamiento de ZFS reflejada se puede clonar fácilmente como copia de seguridad de agrupación mediante el comando `zpool split`.

Actualmente, esta función no puede utilizarse para dividir una agrupación raíz reflejada.

Puede utilizar el comando `zpool dividir` para separar discos desde una agrupación de almacenamiento de ZFS reflejada para crear un nuevo conjunto de conexiones con uno de los discos separado. La nueva agrupación tendrá el mismo contenido que la agrupación original de almacenamiento de ZFS reflejada.

De manera predeterminada, una operación `zpool split` en una agrupación reflejada desvincula el último disco de la agrupación recién creada. Después de la operación de división, importe la nueva agrupación. Por ejemplo:

```
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
```

```
          mirror-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0

errors: No known data errors
# zpool split tank tank2
# zpool import tank2
# zpool status tank tank2
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          c1t0d0    ONLINE       0     0     0

errors: No known data errors

  pool: tank2
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank2       ONLINE       0     0     0
          c1t2d0    ONLINE       0     0     0

errors: No known data errors
```

Puede identificar qué disco utilizar para la nueva agrupación especificando ésta con el comando `zpool split`. Por ejemplo:

```
# zpool split tank tank2 c1t0d0
```

Antes de que se produzca la división, los datos en memoria se vaciarán en los discos reflejados. Después de vaciarse los datos, el disco se desconecta de la agrupación y se le asigna un nuevo GUID de agrupación. Se genera un nuevo GUID para permitir la importación de la agrupación en el mismo sistema en que se ha dividido.

Si la agrupación que se va a dividir tiene puntos de montaje de conjunto de datos no predeterminados y la nueva agrupación se crea en el mismo sistema, tendrá que usar la opción de `zpool split -R` para identificar un directorio raíz alternativo para la nueva, a fin de evitar conflictos entre puntos de montaje. Por ejemplo:

```
# zpool split -R /tank2 tank tank2
```

Si no utiliza la opción de `zpool split -R` y observa que hay conflictos entre puntos de montaje al intentar importar la nueva agrupación, impórtela utilizando la opción `-R`. Si la nueva agrupación se crea en un sistema distinto, no debería ser preciso especificar un directorio raíz alternativo a menos que haya conflictos de puntos de montaje.

Tenga en cuenta lo siguiente antes de utilizar la función `zpool split`:

- Esta función no está disponible para una configuración RAIDZ o una agrupación no redundante de varios discos.
- Antes de intentar una operación `zpool split`, no debería haber activas operaciones de aplicación ni datos.
- Es importante tener discos que respondan al comando de vaciado de caché de escritura del disco, en lugar de pasarlo por alto.
- Una agrupación no se puede dividir si la actualización de duplicación está en curso.
- La división de una agrupación reflejada es óptima cuando se compone de dos o tres discos y el último es la agrupación original utilizada para crear la nueva. Luego puede usar el comando `zpool attach` para volver a crear la agrupación de almacenamiento original reflejada o convertir la agrupación recién creada en agrupación de almacenamiento reflejada. De momento no existe la posibilidad de usar esta función para crear una agrupación reflejada *nueva* a partir de una agrupación reflejada *existente*.
- Si la agrupación ya existente es un reflejo de tres vías, la nueva agrupación contendrá un disco después de la operación de división. Si la agrupación ya existente es un reflejo de dos vías de dos discos, el resultado son dos agrupaciones no redundantes de dos discos. Tendrá que vincular dos discos adicionales con el fin de convertir las agrupaciones no redundantes en agrupaciones reflejadas.
- Una buena forma de mantener los datos redundantes durante una operación de división consiste en dividir una agrupación de almacenamiento reflejada compuesta de tres discos de forma que la agrupación original se componga de dos discos reflejados después de la operación de división.

EJEMPLO 4–7 División de una agrupación de ZFS reflejada

En el ejemplo siguiente se divide una agrupación de almacenamiento reflejada denominada `trinity`, con tres discos, `c1t0d0`, `c1t2d0` y `c1t3d0`. Las dos agrupaciones resultantes son la agrupación reflejada `trinity`, con los discos `c1t0d0` y `c1t2d0`, y la nueva agrupación denominada `neo`, con el disco `c1t3d0`. Cada agrupación tiene el mismo contenido.

```
# zpool status trinity
  pool: trinity
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        trinity     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0

errors: No known data errors
```

EJEMPLO 4–7 División de una agrupación de ZFS reflejada *(Continuación)*

```
# zpool split trinity neo
# zpool import neo
# zpool status trinity neo
  pool: neo
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        neo         ONLINE       0     0     0
          c1t3d0    ONLINE       0     0     0

errors: No known data errors

  pool: trinity
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        trinity     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0

errors: No known data errors
```

Dispositivos con conexión y sin conexión en una agrupación de almacenamiento

ZFS permite que los dispositivos individuales queden sin conexión o con conexión. Cuando el hardware no es fiable o no funciona adecuadamente, ZFS continúa con la lectura o la escritura de datos en el dispositivo, suponiendo que la condición es sólo temporal. Si no es temporal, es posible indicar a ZFS que termine la conexión del dispositivo para que éste se pase por alto. ZFS no envía solicitudes a un dispositivo sin conexión.

Nota – Para sustituir dispositivos no es necesario desconectarlos.

Para desconectar temporalmente el almacenamiento puede utilizar el comando `zpool offline`. Por ejemplo, si tiene que desconectar físicamente una matriz de un conjunto de conmutadores de canal de fibra y conectar la matriz a otro conjunto, puede terminar la conexión de LUN desde la matriz utilizada en las agrupaciones de almacenamiento de ZFS. Con la matriz conectada de nuevo y en funcionamiento en el nuevo conjunto de conmutadores,

puede volver a conectar las mismas LUN. Los datos agregados a las agrupaciones de almacenamiento mientras las LUN estaban sin conexión se actualizarán en las LUN una vez restablecida la conexión.

Esta situación es posible siempre y cuando los sistemas en cuestión detecten el almacenamiento después de conectarlo a los nuevos conmutadores, posiblemente a través de distintos controladores, y las agrupaciones se establecen como configuraciones reflejadas o de RAID-Z.

Cómo terminar la conexión de un dispositivo

Puede terminar la conexión de un dispositivo mediante el comando `zpool offline`. El dispositivo se puede especificar mediante la ruta o un nombre abreviado, si el dispositivo es un disco. Por ejemplo:

```
# zpool offline tank c1t0d0
bringing device c1t0d0 offline
```

Tenga en cuenta los puntos siguientes al desconectar un dispositivo:

- Una agrupación no se puede desconectar si genera errores. Por ejemplo, no puede desconectar dos dispositivos de una configuración `raidz1`, ni tampoco puede desconectar un dispositivo virtual de nivel superior.

  ```
  # zpool offline tank c1t0d0
  cannot offline c1t0d0: no valid replicas
  ```

- De modo predeterminado, el estado `OFFLINE` es persistente. El dispositivo permanece sin conexión cuando el sistema se reinicia.

  Para desconectar temporalmente un dispositivo, utilice la opción `zpool offline -t`. Por ejemplo:

  ```
  # zpool offline -t tank c1t0d0
   bringing device 'c1t0d0' offline
  ```

  Cuando el sistema se reinicia, este dispositivo vuelve automáticamente al estado `ONLINE`.

- Si un dispositivo se queda sin conexión, no se desconecta de la agrupación de almacenamiento. Si intenta utilizar el dispositivo sin conexión en otra agrupación, incluso después de que la agrupación original se haya destruido, aparece en pantalla un mensaje similar al siguiente:

  device `is part of exported or potentially active ZFS` *pool*`. Please see zpool(1M)`

  Si desea utilizar el dispositivo sin conexión en otra agrupación de almacenamiento después de destruir la agrupación de almacenamiento original, conecte el dispositivo y destruya la agrupación de almacenamiento original.

  Otra forma de utilizar un dispositivo de otra agrupación de almacenamiento a la vez que se mantiene la agrupación de almacenamiento original consiste en sustituir el dispositivo de la

agrupación de almacenamiento original por otro equivalente. Para obtener información sobre la sustitución de dispositivos, consulte "Sustitución de dispositivos en una agrupación de almacenamiento" en la página 96.

Los dispositivos sin conexión aparecen con el estado `OFFLINE` al consultar el estado de la agrupación. Para obtener información sobre cómo saber el estado de la agrupación, consulte "Consulta del estado de una agrupación de almacenamiento de ZFS" en la página 107.

Para obtener más información sobre la situación del dispositivo, consulte "Cómo determinar el estado de las agrupaciones de almacenamiento de ZFS" en la página 113.

Cómo conectar un dispositivo

Si se anula la conexión de un dispositivo, se puede restablecer mediante el comando `zpool online`. Por ejemplo:

```
# zpool online tank c1t0d0
bringing device c1t0d0 online
```

Si se conecta un dispositivo, los datos escritos en la agrupación se vuelven a sincronizar con el dispositivo que acaba de quedar disponible. Para sustituir un disco no se puede utilizar un dispositivo con conexión. Si desconecta un dispositivo, reemplaza el dispositivo e intenta conectarlo, queda en estado de error.

Si intenta conectar un dispositivo defectuoso, aparece un mensaje similar al siguiente:

```
# zpool online tank c1t0d0
warning: device 'c1t0d0' onlined, but remains in faulted state
use 'zpool replace' to replace devices that are no longer present
```

También puede que vea el mensaje de disco defectuoso en la consola o escrito en el archivo `/var/adm/messages`. Por ejemplo:

```
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Wed Jun 30 14:53:39 MDT 2010
PLATFORM: SUNW,Sun-Fire-880, CSN: -, HOSTNAME: neo
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: 504a1188-b270-4ab0-af4e-8a77680576b8
DESC: A ZFS device failed.  Refer to http://sun.com/msg/ZFS-8000-D3 for more information.
AUTO-RESPONSE: No automated response will occur.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.
```

Para obtener más información sobre cómo reemplazar un dispositivo defectuoso, consulte "Resolución de un dispositivo que no se encuentra" en la página 303.

Puede utilizar el comando `zpool online -e` para expandir una LUN. De manera predeterminada, una LUN que se agrega a una agrupación no se expande a su tamaño máximo a

menos que esté activada la propiedad de agrupación `autoexpand`. Puede ampliar la LUN automáticamente por medio del comando `zpool online -e` con la LUN tanto conectada como sin conexión. Por ejemplo:

```
# zpool online -e tank c1t13d0
```

Borrado de errores de dispositivo de agrupación de almacenamiento

Si un dispositivo pierde la conexión por un fallo que hace que los errores aparezcan en la salida de `zpool status`, los recuentos de errores se pueden borrar con el comando `zpool clear`.

Si se especifica sin argumentos, este comando borra todos los errores de dispositivo de la agrupación. Por ejemplo:

```
# zpool clear tank
```

Si se especifican uno o más dispositivos, este comando sólo borra errores asociados con los dispositivos especificados. Por ejemplo:

```
# zpool clear tank c1t0d0
```

Para obtener más información sobre cómo borrar errores de `zpool`, consulte "Supresión de errores transitorios" en la página 307.

Sustitución de dispositivos en una agrupación de almacenamiento

Puede reemplazar un dispositivo en una agrupación de almacenamiento mediante el comando `zpool replace`.

Si se reemplaza físicamente un dispositivo por otro en la misma ubicación de una agrupación redundante, puede que sólo haga falta identificar el dispositivo sustituido. ZFS reconoce que el dispositivo es un disco diferente en la misma ubicación de cierto hardware. Por ejemplo, para reemplazar un disco defectuoso (`c1t1d0`) quitándolo y colocándolo en la misma ubicación, emplee la siguiente sintaxis:

```
# zpool replace tank c1t1d0
```

Si va a reemplazar un dispositivo de una agrupación de almacenamiento con un disco en otra ubicación física, tendrá que especificar ambos dispositivos. Por ejemplo:

```
# zpool replace tank c1t1d0 c1t2d0
```

Si desea reemplazar un disco en la agrupación de almacenamiento de ZFS, consulte "Cómo sustituir un disco en la agrupación raíz ZFS" en la página 178.

A continuación se detalla el procedimiento básico para sustituir un disco:

- Si es preciso, desconecte el dispositivo con el comando `zpool offline`.
- Retire el disco que se debe reemplazar.
- Inserte el disco nuevo.
- Ejecute el comando `zpool replace`. Por ejemplo:

  ```
  # zpool replace tank c1t1d0
  ```
- Conecte el disco mediante el comando `zpool online`.

En sistemas como Sun Fire x4500, antes de desconectar un disco se debe desconfigurar. Si va a reemplazar un disco en la misma posición de ranura en este sistema, puede ejecutar el comando `zpool replace` del modo descrito en el primer ejemplo de esta sección.

Para ver un ejemplo de sustitución de un disco en un sistema Sun Fire X4500, consulte el Ejemplo 11–1.

Tenga en cuenta lo siguiente al sustituir dispositivos en una agrupación de almacenamiento de ZFS:

- Si la propiedad de agrupación `autoreplace` se configura como habilitada `(on)`, se aplicará formato y sustitución a cualquier dispositivo que se encuentre en la misma ubicación física que un dispositivo previamente perteneciente a la ubicación. No es necesario que utilice el comando `zpool replace` cuando esta propiedad está habilitada. Es posible que no todos los tipos de hardware dispongan de esta función.
- El tamaño del dispositivo de sustitución debe ser igual o mayor que el disco más pequeño en una configuración de RAID-Z o reflejada.
- Cuando se agrega a una agrupación un dispositivo de sustitución mayor que el dispositivo al que va a sustituir, no se amplía automáticamente a su tamaño máximo. El valor de la propiedad `autoexpand` determina si una LUN de sustitución se amplía a su tamaño máximo cuando el disco se agrega a la agrupación. De manera predeterminada, la propiedad `autoexpand` está habilitada. Se puede habilitar esta propiedad para ampliar el tamaño de LUN antes o después de agregar a la agrupación la LUN mayor.

  En el ejemplo siguiente, se sustituyen dos discos de 16 GB de una agrupación reflejado por dos discos de 72 GB. La propiedad `autoexpand` está habilitada tras las sustituciones de disco para ampliar el tamaño de LUN.

  ```
  # zpool create pool mirror c1t16d0 c1t17d0
  # zpool status
    pool: pool
   state: ONLINE
  ```

```
 scrub: none requested
config:

        NAME         STATE     READ WRITE CKSUM
        pool         ONLINE       0     0     0
          mirror     ONLINE       0     0     0
            c1t16d0  ONLINE       0     0     0
            c1t17d0  ONLINE       0     0     0

zpool list pool
NAME   SIZE   ALLOC  FREE    CAP  HEALTH  ALTROOT
pool  16.8G  76.5K  16.7G     0%  ONLINE  -
# zpool replace pool c1t16d0 c1t1d0
# zpool replace pool c1t17d0 c1t2d0
# zpool list pool
NAME   SIZE   ALLOC  FREE    CAP  HEALTH  ALTROOT
pool  16.8G  88.5K  16.7G     0%  ONLINE  -
# zpool set autoexpand=on pool
# zpool list pool
NAME   SIZE   ALLOC  FREE    CAP  HEALTH  ALTROOT
pool  68.2G   117K  68.2G     0%  ONLINE  -
```

- La sustitución de muchos discos en una agrupación de gran tamaño tarda mucho en realizarse debido al proceso de actualizar la duplicación de los datos en los discos nuevos. Además, es recomendable ejecutar el comando `zpool scrub` entre sustituciones de discos, para asegurarse de que los dispositivos de sustitución estén operativos y que los datos se escriban correctamente.
- Si se ha sustituido automáticamente un disco fallido con un repuesto en marcha, es posible que sea necesario desconectar el repuesto después de sustituir el disco fallido. Para obtener información sobre cómo desconectar un repuesto en marcha, consulte "Activación y desactivación de repuestos en marcha en la agrupación de almacenamiento" en la página 100.

Para obtener más información sobre cómo reemplazar dispositivos, consulte "Resolución de un dispositivo que no se encuentra" en la página 303 y "Sustitución o reparación de un dispositivo dañado" en la página 305.

Designación de repuestos en marcha en la agrupación de almacenamiento

La función de repuesto en marcha permite identificar discos que se podrían utilizar para sustituir un dispositivo defectuoso o en estado "faulted" en una o más agrupaciones de almacenamiento. Si un dispositivo se designa como *repuesto en marcha*, significa que no es un dispositivo activo en una agrupación. Ahora bien, si un dispositivo activo falla, el repuesto en marcha sustituye automáticamente al defectuoso.

Los dispositivos se pueden designar como repuestos en marcha de los modos siguientes:

- Cuando se crea la agrupación con el comando `zpool create`.
- Después de crear la agrupación con el comando `zpool add`.
- Los dispositivos de repuestos en marcha se pueden compartir entre varias agrupaciones, pero no si las agrupaciones están en distintos sistemas.

El ejemplo siguiente muestra cómo designar dispositivos como repuestos en marcha cuando se crea la agrupación:

```
# zpool create trinity mirror c1t1d0 c2t1d0 spare c1t2d0 c2t2d0
# zpool status trinity
  pool: trinity
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        trinity     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0
        spares
          c1t2d0    AVAIL
          c2t2d0    AVAIL

errors: No known data errors
```

El ejemplo siguiente muestra cómo designar repuestos en marcha agregándolos a una agrupación después de crearla:

```
# zpool add neo spare c5t3d0 c6t3d0
# zpool status neo
  pool: neo
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        neo         ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c3t3d0  ONLINE       0     0     0
            c4t3d0  ONLINE       0     0     0
        spares
          c5t3d0    AVAIL
          c6t3d0    AVAIL

errors: No known data errors
```

Los repuestos en marcha se pueden suprimir de una agrupación de almacenamiento mediante el comando `zpool remove`. Por ejemplo:

```
# zpool remove zeepool c2t3d0
# zpool status zeepool
  pool: zeepool
```

```
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        zeepool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0
        spares
          c1t3d0    AVAIL

errors: No known data errors
```

No se puede suprimir un repuesto en marcha si se está utilizando en una agrupación de almacenamiento.

Tenga en cuenta lo siguiente al utilizar repuestos en marcha de ZFS:

- Actualmente, el comando `zpool remove` sólo es apto para eliminar repuestos en marcha, dispositivos caché y dispositivos de registro.
- Para agregar un disco como repuesto en marcha, el repuesto en marcha debe ser igual o mayor que el disco más grande de la agrupación. Se puede agregar un disco de repuesto de tamaño inferior. Ahora bien, al activar ese disco de repuesto de tamaño inferior, de forma automática o con el comando `zpool replace`, la operación falla y genera un mensaje de error parecido al siguiente:

  `cannot replace` *disk3* `with` *disk4*`: device is too small`

Activación y desactivación de repuestos en marcha en la agrupación de almacenamiento

Los repuestos en marcha se activan de los modos siguientes:

- Sustitución manual: sustituya un dispositivo incorrecto en una agrupación de almacenamiento con un repuesto en marcha mediante el comando `zpool replace`.
- Sustitución automática: cuando se detecta un error, un agente FMA examina la agrupación para ver si dispone de repuestos en marcha. Si es así, sustituye el dispositivo con errores por un repuesto en marcha.

  Si falla un repuesto en marcha que está en uso, el agente FMA quita el repuesto y cancela la sustitución. El agente intenta sustituir el dispositivo por otro repuesto en marcha, si lo hay. Esta función está limitada por el hecho de que el motor de diagnóstico ZFS sólo emite errores cuando un dispositivo desaparece del sistema.

  Si sustituye físicamente un dispositivo defectuoso con un repuesto activo, puede reactivar el original, pero debe desactivar el dispositivo reemplazado mediante el comando `zpool detach` para desconectar el repuesto. Si configura la propiedad de agrupación `autoreplace` como habilitada `(on)`, el repuesto se desconecta automáticamente y vuelve a la agrupación de repuestos cuando se inserta el dispositivo nuevo y se completa la operación de conexión.

Puede sustituir manualmente un dispositivo con un repuesto en marcha mediante el comando `zpool replace`. Consulte el Ejemplo 4–8.

Un dispositivo con errores se sustituye automáticamente si hay un repuesto en marcha. Por ejemplo:

```
# zpool status -x
  pool: zeepool
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: resilver completed after 0h0m with 0 errors on Mon Jan 11 10:20:35 2010
config:

        NAME          STATE     READ WRITE CKSUM
        zeepool       DEGRADED     0     0     0
          mirror-0    DEGRADED     0     0     0
            c1t2d0    ONLINE       0     0     0
            spare-1   DEGRADED     0     0     0
              c2t1d0  UNAVAIL      0     0     0  cannot open
              c2t3d0  ONLINE       0     0     0  88.5K resilvered
        spares
          c2t3d0      INUSE     currently in use

errors: No known data errors
```

Actualmente se puede desactivar un repuesto en marcha de las siguientes maneras:

- Eliminando el repuesto de la agrupación de almacenamiento.
- Desconectando el repuesto después de la sustitución de un disco fallido. Consulte el Ejemplo 4–9.
- Cambiando el repuesto de forma temporal o permanente. Consulte el Ejemplo 4–10.

EJEMPLO 4–8 Sustitución manual de un disco con un repuesto en marcha

En este ejemplo, el comando `zpool replace` se utiliza para sustituir el disco `c2t1d0` con el repuesto en marcha `c2t3d0`.

```
# zpool replace zeepool c2t1d0 c2t3d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 10:00:50 2010
config:

        NAME          STATE     READ WRITE CKSUM
        zeepool       ONLINE       0     0     0
          mirror-0    ONLINE       0     0     0
            c1t2d0    ONLINE       0     0     0
            spare-1   ONLINE       0     0     0
              c2t1d0  ONLINE       0     0     0
```

EJEMPLO 4–8 Sustitución manual de un disco con un repuesto en marcha *(Continuación)*

```
            c2t3d0  ONLINE       0     0     0  90K resilvered
        spares
          c2t3d0      INUSE     currently in use

errors: No known data errors
```

A continuación, quite el disco `c2t1d0`.

```
# zpool detach zeepool c2t1d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 10:00:50 2010
config:

        NAME        STATE     READ WRITE CKSUM
        zeepool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0  90K resilvered

errors: No known data errors
```

EJEMPLO 4–9 Desconexión de un repuesto en marcha después de sustituir el disco fallido

En este ejemplo, el disco averiado (`c2t1d0`) se sustituye físicamente y ZFS recibe una notificación mediante el comando `zpool replace`.

```
# zpool replace zeepool c2t1d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 10:08:44 2010
config:

        NAME          STATE     READ WRITE CKSUM
        zeepool       ONLINE       0     0     0
          mirror-0    ONLINE       0     0     0
            c1t2d0    ONLINE       0     0     0
            spare-1   ONLINE       0     0     0
              c2t3d0  ONLINE       0     0     0  90K resilvered
              c2t1d0  ONLINE       0     0     0
        spares
          c2t3d0      INUSE     currently in use

errors: No known data errors
```

A continuación se puede utilizar el comando `zpool detach` para volver a dejar el repuesto en marcha en la agrupación de repuestos. Por ejemplo:

```
# zpool detach zeepool c2t3d0
# zpool status zeepool
```

EJEMPLO 4–9 Desconexión de un repuesto en marcha después de sustituir el disco fallido *(Continuación)*

```
  pool: zeepool
 state: ONLINE
 scrub: resilver completed with 0 errors on Wed Jan 20 10:08:44 2010
config:

        NAME              STATE     READ WRITE CKSUM
        zeepool           ONLINE       0     0     0
          mirror          ONLINE       0     0     0
            c1t2d0        ONLINE       0     0     0
            c2t1d0        ONLINE       0     0     0
        spares
          c2t3d0          AVAIL

errors: No known data errors
```

EJEMPLO 4–10 Desconexión de un disco averiado y uso del repuesto en marcha

Si desea sustituir un disco fallido mediante un intercambio temporal o permanente del repuesto en marcha que lo está sustituyendo, desconecte el disco original (fallido). Si se sustituye el disco fallido en algún momento, se podrá agregar a la agrupación de almacenamiento como repuesto. Por ejemplo:

```
# zpool status zeepool
  pool: zeepool
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: resilver in progress for 0h0m, 70.47% done, 0h0m to go
config:

        NAME          STATE     READ WRITE CKSUM
        zeepool       DEGRADED     0     0     0
          mirror-0    DEGRADED     0     0     0
            c1t2d0    ONLINE       0     0     0
            spare-1   DEGRADED     0     0     0
              c2t1d0  UNAVAIL      0     0     0  cannot open
              c2t3d0  ONLINE       0     0     0  70.5M resilvered
        spares
          c2t3d0      INUSE     currently in use

errors: No known data errors
# zpool detach zeepool c2t1d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 13:46:46 2010
config:

        NAME        STATE     READ WRITE CKSUM
        zeepool     ONLINE       0     0     0
```

EJEMPLO 4–10 Desconexión de un disco averiado y uso del repuesto en marcha *(Continuación)*

```
          mirror-0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0  70.5M resilvered

errors: No known data errors
(Original failed disk c2t1d0 is physically replaced)
# zpool add zeepool spare c2t1d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 13:48:46 2010
config:

        NAME        STATE     READ WRITE CKSUM
        zeepool     ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0  70.5M resilvered
        spares
          c2t1d0    AVAIL

errors: No known data errors
```

Administración de propiedades de agrupaciones de almacenamiento de ZFS

Utilice el comando `zpool get` para visualizar información de propiedades de almacenamiento. Por ejemplo:

```
# zpool get all mpool
NAME  PROPERTY       VALUE       SOURCE
pool  size           68G         -
pool  capacity       0%          -
pool  altroot        -           default
pool  health         ONLINE      -
pool  guid           601891032394735745  default
pool  version        22          default
pool  bootfs         -           default
pool  delegation     on          default
pool  autoreplace    off         default
pool  cachefile      -           default
pool  failmode       wait        default
pool  listsnapshots  on          default
pool  autoexpand     off         default
pool  free           68.0G       -
pool  allocated      76.5K       -
```

Con el comando `zpool set` se pueden establecer las propiedades de agrupaciones de almacenamiento. Por ejemplo:

```
# zpool set autoreplace=on mpool
# zpool get autoreplace mpool
NAME  PROPERTY     VALUE    SOURCE
mpool autoreplace  on       default
```

TABLA 4–1 Descripciones de propiedades de agrupaciones ZFS

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`allocated`	Cadena	N/D	Valor de sólo lectura que identifica la cantidad de espacio de almacenamiento dentro de la agrupación que se ha asignado físicamente.
`altroot`	Cadena	`off`	Identifica un directorio raíz alternativo. Si se establece, el directorio se antepone a cualquier punto de montaje de la agrupación. Esta propiedad es apta para examinar una agrupación desconocida, si los puntos de montaje no son de confianza o en un entorno de inicio alternativo en que las rutas habituales no sean válidas.
`autoreplace`	Booleano	`off`	Controla el reemplazo automático de dispositivos. Si la propiedad se establece en `off`, la sustitución del dispositivo debe iniciarla el administrador mediante el comando `zpool replace`. Si se establece en `on`, automáticamente se da formato y se sustituye cualquier dispositivo nuevo que se detecte en la misma ubicación física que un dispositivo previamente perteneciente a la agrupación. La abreviatura de la propiedad es `replace`.
`bootfs`	Booleano	N/D	Identifica el conjunto de datos predeterminado que se puede iniciar para la agrupación raíz. Esta propiedad la suelen establecer los programas de instalación y actualización.
`cachefile`	Cadena	N/D	Los controles donde se almacena en la memoria caché la configuración de agrupación. Todas las agrupaciones de la caché se importan automáticamente cuando se inicia el sistema. Sin embargo, los entornos de instalación y administración de clústeres podrían requerir el almacenamiento en caché de esta información en otra ubicación, para impedir la importación automática de las agrupaciones. Puede configurar esta propiedad para almacenar en caché información de configuración de agrupación en una ubicación distinta. Esta información se puede importar más adelante mediante el comando `zpool import -c`. La mayoría de las configuraciones ZFS no usan esta propiedad.
`capacity`	Número	N/D	Valor de sólo lectura que identifica el porcentaje del espacio de agrupación utilizado. La abreviatura de la propiedad es `cap`.

TABLA 4–1 Descripciones de propiedades de agrupaciones ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`delegation`	Booleano	`on`	Controla si a un usuario sin privilegios se le pueden conceder permisos de acceso que se definen para un conjunto de datos. Para más información, consulte el Capítulo 9, "Administración delegada de ZFS".
`failmode`	Cadena	`wait`	Controla el comportamiento del sistema en caso de producirse un error grave de agrupación. Esta situación suele deberse a la pérdida de conexión con el dispositivo o los dispositivos de almacenamiento subyacentes, o a un error de todos los dispositivos de la agrupación. El comportamiento de dicho evento depende de uno de estos valores: ■ `wait`: bloquea todas las solicitudes de acceso a la agrupación hasta que se restablece la conexión del dispositivo y los errores se borran mediante el comando `zpool clear`. En este estado, las operaciones de E/S de la agrupación están bloqueadas, pero las operaciones de lectura podrían ser viables. Una agrupación se mantiene en estado `wait` hasta que se resuelve el problema del dispositivo. ■ `continue`: devuelve un error EIO a cualquier solicitud de E/S nueva, pero permite lecturas en cualquier otro dispositivo que esté en buen estado. Se bloqueará cualquier solicitud pendiente de ejecución en el disco. Después de volver a colocar o conectar el dispositivo, los errores se deben eliminar con el comando `zpool clear`. ■ `panic`: imprime un mensaje en la consola y genera un volcado de bloqueo del sistema.
`free`	Cadena	N/D	Valor de sólo lectura que identifica el número de bloques aún sin asignar dentro de la agrupación.
`guid`	Cadena	N/D	Propiedad de sólo lectura que detecta el identificador exclusivo de la agrupación.
`health`	Cadena	N/D	Propiedad de sólo lectura que identifica el estado actual de la agrupación y lo establece en ONLINE, DEGRADED, FAULTED, OFFLINE, REMOVED o UNAVAIL
`listsnapshots`	Cadena	`on`	Controla si la información de instantánea que está asociada con esta agrupación se muestra con el comando `zfs list`. Si esta propiedad está inhabilitada, la información de la instantánea se puede mostrar con el comando `zfs list -t snapshot`.

TABLA 4–1 Descripciones de propiedades de agrupaciones ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`size`	Número	N/D	Propiedad de sólo lectura que identifica el tamaño total de la agrupación de almacenamiento.
`version`	Número	N/D	Identifica la versión actual en disco de la agrupación. El método preferido de actualizar agrupaciones se realiza con el comando `zpool upgrade`, aunque esta propiedad se puede utilizar si se necesita una versión predeterminada por cuestiones de compatibilidad retroactiva. Esta propiedad se puede establecer en cualquier número que esté entre 1 y la versión actual indicada por el comando `zpool upgrade -v`.

Consulta del estado de una agrupación de almacenamiento de ZFS

El comando `zpool list` ofrece diversos modos de solicitar información sobre el estado de la agrupación. La información disponible suele pertenecer a una de estas tres categorías: información básica de utilización, estadística de E/S y situación. En esta sección se abordan los tres tipos de información de agrupaciones de almacenamiento.

- "Visualización de información de agrupaciones de almacenamiento de ZFS" en la página 107
- "Visualización de estadísticas de E/S de agrupaciones de almacenamiento de ZFS " en la página 111
- "Cómo determinar el estado de las agrupaciones de almacenamiento de ZFS" en la página 113

Visualización de información de agrupaciones de almacenamiento de ZFS

El comando `zpool list` es apto para mostrar información básica sobre agrupaciones.

Visualización de información relativa a todas las agrupaciones de almacenamiento o a uno concreto

Sin argumentos, el comando `zpool list` sólo muestra los siguientes datos para todas las agrupaciones del sistema:

```
# zpool list
NAME                    SIZE    ALLOC   FREE    CAP  HEALTH     ALTROOT
tank                   80.0G   22.3G   47.7G    28%  ONLINE     -
dozer                   1.2T    384G    816G    32%  ONLINE     -
```

La salida de este comando muestra lo siguientes datos:

NAME	El nombre de la agrupación.
SIZE	El tamaño total de la agrupación, igual a la suma del tamaño de todos los dispositivos virtuales de nivel superior.
ALLOC	La cantidad de espacio físico asignada a todos los conjuntos de datos y los metadatos internos. Esta cantidad es diferente de la cantidad de espacio en el disco según se indica en el nivel del sistema de archivos. Para obtener más información sobre la especificación del espacio disponible en el sistema de archivos, consulte "Cálculo del espacio de ZFS" en la página 62.
FREE	Cantidad de espacio sin asignar en la agrupación.
CAP (CAPACITY)	Cantidad de espacio utilizado, expresada como porcentaje del espacio total en el disco.
HEALTH	Estado actual de la agrupación. Para obtener más información sobre la situación de la agrupación, consulte "Cómo determinar el estado de las agrupaciones de almacenamiento de ZFS" en la página 113.
ALTROOT	Raíz alternativa de la agrupación, de haberla. Para obtener más información sobre las agrupaciones raíz alternativas, consulte "Uso de agrupaciones raíz de ZFS alternativas" en la página 289.

También puede reunir estadísticas para una agrupación determinada especificando el nombre de la agrupación. Por ejemplo:

```
# zpool list tank
NAME                    SIZE    ALLOC   FREE    CAP   HEALTH     ALTROOT
tank                   80.0G    22.3G   47.7G    28%  ONLINE     -
```

Visualización de estadísticas de una agrupación de almacenamiento específico

Las estadísticas específicas se pueden solicitar mediante la opción `-o`. Esta opción ofrece informes personalizados o un modo rápido de visualizar la información pertinente. Por ejemplo, para ver sólo el nombre y el tamaño de cada agrupación, utilice la sintaxis siguiente:

```
# zpool list -o name,size
NAME                     SIZE
tank                    80.0G
dozer                    1.2T
```

Los nombres de columna corresponden a las propiedades que se enumeran en "Visualización de información relativa a todas las agrupaciones de almacenamiento o a uno concreto" en la página 107.

Salida de la secuencia de comandos de la agrupación de almacenamiento de ZFS

La salida predeterminada del comando `zpool list` está diseñada para mejorar la legibilidad; no es fácil de utilizar como parte de una secuencia de comandos shell. Para facilitar los usos de programación del comando, la opción `-H` es válida para suprimir encabezados de columna y separar los campos con tabuladores, en lugar de espacios. Por ejemplo, para solicitar una lista con todos los nombres de agrupaciones en el sistema debería usar esta sintaxis:

```
# zpool list -Ho name
tank
dozer
```

Aquí puede ver otro ejemplo:

```
# zpool list -H -o name,size
tank   80.0G
dozer  1.2T
```

Cómo mostrar el historial de comandos de la agrupación de almacenamiento de ZFS

ZFS registra automáticamente los comandos `zfs` y `zpool` que se ejecutan satisfactoriamente para modificar la información de estado de la agrupación. Esta información se puede mostrar mediante el comando `zpool history`.

Por ejemplo, la sintaxis siguiente muestra la salida del comando para la agrupación raíz:

```
# zpool history
History for 'rpool':
2010-05-11.10:18:54 zpool create -f -o failmode=continue -R /a -m legacy -o
cachefile=/tmp/root/etc/zfs/zpool.cache rpool mirror c1t0d0s0 c1t1d0s0
2010-05-11.10:18:55 zfs set canmount=noauto rpool
2010-05-11.10:18:55 zfs set mountpoint=/rpool rpool
2010-05-11.10:18:56 zfs create -o mountpoint=legacy rpool/ROOT
2010-05-11.10:18:57 zfs create -b 8192 -V 2048m rpool/swap
2010-05-11.10:18:58 zfs create -b 131072 -V 1536m rpool/dump
2010-05-11.10:19:01 zfs create -o canmount=noauto rpool/ROOT/zfsBE
2010-05-11.10:19:02 zpool set bootfs=rpool/ROOT/zfsBE rpool
2010-05-11.10:19:02 zfs set mountpoint=/ rpool/ROOT/zfsBE
2010-05-11.10:19:03 zfs set canmount=on rpool
2010-05-11.10:19:04 zfs create -o mountpoint=/export rpool/export
2010-05-11.10:19:05 zfs create rpool/export/home
2010-05-11.11:11:10 zpool set bootfs=rpool rpool
2010-05-11.11:11:10 zpool set bootfs=rpool/ROOT/zfsBE rpool
```

Puede utilizar una salida similar en el sistema para identificar el conjunto *exacto* de comandos de ZFS que se han ejecutado para resolver una situación de error.

Este registro de historial presenta las características siguientes:

- El registro no se puede inhabilitar.
- El registro se mantiene de forma persistente en el disco, lo que significa que se guarda en los reinicios del sistema.
- El registro se implementa como búfer de anillo. El tamaño mínimo es de 128 KB. El tamaño máximo es de 32 MB.
- En agrupaciones pequeñas, el tamaño máximo se restringe al 1% del tamaño de la agrupación, donde el *tamaño* se determina al crear agrupaciones.
- El registro no requiere administración; eso significa que no es necesario ajustar el tamaño del registro ni cambiar la ubicación del registro.

Para identificar el historial de comandos de una agrupación de almacenamiento específica, utilice una sintaxis similar a la siguiente:

```
# zpool history tank
History for 'tank':
2010-05-13.14:13:15 zpool create tank mirror c1t2d0 c1t3d0
2010-05-13.14:21:19 zfs create tank/snaps
2010-05-14.08:10:29 zfs create tank/ws01
2010-05-14.08:10:54 zfs snapshot tank/ws01@now
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix
```

Utilice la opción `-l` para ver el formato completo que incluye el nombre de usuario, el nombre de host y la zona en que se ha efectuado la operación. Por ejemplo:

```
# zpool history -l tank
History for 'tank':
2010-05-13.14:13:15 zpool create tank mirror c1t2d0 c1t3d0 [user root on neo]
2010-05-13.14:21:19 zfs create tank/snaps [user root on neo]
2010-05-14.08:10:29 zfs create tank/ws01 [user root on neo]
2010-05-14.08:10:54 zfs snapshot tank/ws01@now [user root on neo]
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix [user root on neo]
```

Utilice la opción `-i` para ver información de eventos internos válida para tareas de diagnóstico. Por ejemplo:

```
# zpool history -i tank
2010-05-13.14:13:15 zpool create -f tank mirror c1t2d0 c1t23d0
2010-05-13.14:13:45 [internal pool create txg:6] pool spa 19; zfs spa 19; zpl 4;...
2010-05-13.14:21:19 zfs create tank/snaps
2010-05-13.14:22:02 [internal replay_inc_sync txg:20451] dataset = 41
2010-05-13.14:25:25 [internal snapshot txg:20480] dataset = 52
2010-05-13.14:25:25 [internal destroy_begin_sync txg:20481] dataset = 41
2010-05-13.14:25:26 [internal destroy txg:20488] dataset = 41
2010-05-13.14:25:26 [internal reservation set txg:20488] 0 dataset = 0
```

```
2010-05-14.08:10:29 zfs create tank/ws01
2010-05-14.08:10:54 [internal snapshot txg:53992] dataset = 42
2010-05-14.08:10:54 zfs snapshot tank/ws01@now
2010-05-14.08:11:04 [internal create txg:53994] dataset = 58
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix
```

Visualización de estadísticas de E/S de agrupaciones de almacenamiento de ZFS

Para solicitar estadísticas de E/S relativas a agrupaciones o dispositivos virtuales específicos, utilice el comando `zpool iostat`. Similar al comando `iostat`, este comando puede mostrar una instantánea estática de toda la actividad de E/S, así como las estadísticas actualizadas para cada intervalo especificado. Se informa de las estadísticas siguientes:

`alloc capacity` — Cantidad de datos almacenados en la agrupación o el dispositivo. Esta cifra difiere de la cantidad de espacio disponible en los sistemas de archivos reales en una pequeña cantidad debido a detalles de implementación internos.

Para obtener más información sobre la diferencia entre el espacio de la agrupación y el del conjunto de datos, consulte "Cálculo del espacio de ZFS" en la página 62.

`free capacity` — Cantidad de espacio en el disco disponible en la agrupación o dispositivo. Al igual que con la estadística `used`, esta cantidad difiere por un pequeño margen de la cantidad de espacio en el disco disponible para conjuntos de datos.

`read operations` — Número de operaciones de E/S de lectura enviadas a la agrupación o al dispositivo, incluidas las solicitudes de metadatos.

`write operations` — Número de operaciones de E/S de escritura enviadas a la agrupación o al dispositivo.

`read bandwidth` — Ancho de banda de todas las operaciones de lectura (incluidos los metadatos), expresado en unidades por segundo.

`write bandwidth` — Ancho de banda de todas las operaciones de escritura, expresadas en unidades por segundo.

Lista de estadísticas de E/S de todas las agrupaciones

Sin opciones, el comando `zpool iostat` muestra las estadísticas acumuladas desde el inicio de todas las agrupaciones del sistema. Por ejemplo:

```
# zpool iostat
               capacity     operations    bandwidth
```

```
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
rpool       6.05G  61.9G      0      0    786    107
tank        31.3G  36.7G      4      1   296K  86.1K
----------  -----  -----  -----  -----  -----  -----
```

Como estas estadísticas se acumulan desde el inicio, el ancho de banda puede parecer bajo si la agrupación está relativamente inactiva. Para solicitar una vista más exacta del uso actual del ancho de banda, especifique un intervalo. Por ejemplo:

```
# zpool iostat tank 2
               capacity     operations    bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
tank        18.5G  49.5G      0    187      0  23.3M
tank        18.5G  49.5G      0    464      0  57.7M
tank        18.5G  49.5G      0    457      0  56.6M
tank        18.8G  49.2G      0    435      0  51.3M
```

En este ejemplo, el comando muestra las estadísticas de uso de la agrupación `tank` cada dos segundos hasta que se pulsa Ctrl-C. Otra opción consiste en especificar un parámetro `count` adicional con el que el comando se termina tras el número especificado de iteraciones. Por ejemplo, `zpool iostat 2 3` imprimiría un resumen cada dos segundos para tres iteraciones, durante un total de seis segundos. Si sólo hay una agrupación, las estadísticas se muestran en líneas consecutivas. Si hay más de una agrupación, la línea de guiones adicional marca cada iteración para ofrecer una separación visual.

Lista de estadísticas de E/S de dispositivos virtuales

Además de las estadísticas de E/S de todas las agrupaciones, el comando `zpool iostat` puede mostrar estadísticas de E/S para dispositivos virtuales. Este comando se puede usar para identificar dispositivos anormalmente lentos o para observar la distribución de E/S generada por ZFS. Para solicitar toda la distribución de dispositivos virtuales, así como todas las estadísticas de E/S, utilice el comando `zpool iostat -v`. Por ejemplo:

```
# zpool iostat -v
               capacity     operations    bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
rpool       6.05G  61.9G      0      0    785    107
  mirror    6.05G  61.9G      0      0    785    107
    c1t0d0s0    -      -      0      0    578    109
    c1t1d0s0    -      -      0      0    595    109
----------  -----  -----  -----  -----  -----  -----
tank        36.5G  31.5G      4      1   295K   146K
  mirror    36.5G  31.5G    126     45  8.13M  4.01M
    c1t2d0      -      -      0      3   100K   386K
    c1t3d0      -      -      0      3   104K   386K
----------  -----  -----  -----  -----  -----  -----
```

Tenga en cuenta dos puntos importantes al visualizar estadísticas de E/S de dispositivos virtuales:

- En primer lugar, las estadísticas de uso del espacio en el disco sólo están disponibles para dispositivos virtuales de nivel superior. El modo en que el espacio en el disco se asigna entre el reflejo y los dispositivos virtuales RAID-Z es específico de la implementación y es difícil de expresar en un solo número.
- Segundo, los números quizá no se agreguen exactamente como cabría esperar. En concreto, las operaciones en dispositivos reflejados y RAID-Z no serán exactamente iguales. Esta diferencia se aprecia sobre todo inmediatamente después de crear una agrupación, puesto que una cantidad significativa de E/S se efectúa directamente en los discos como parte de la creación de agrupaciones y no se tiene en cuenta en el nivel del reflejo. Con el tiempo se igualan estos números. Pero esta simetría se puede ver afectada si hay dispositivos defectuosos, averiados o desconectados.

Puede utilizar el mismo conjunto de opciones (interval y count) al examinar estadísticas de dispositivos virtuales.

Cómo determinar el estado de las agrupaciones de almacenamiento de ZFS

ZFS ofrece un método integrado para examinar el estado de dispositivos y agrupaciones. La situación de una agrupación la determina el estado de todos sus dispositivos. Esta información sobre el estado se obtiene con el comando `zpool status`. Además, `fmd` informa de posibles errores en dispositivos y agrupaciones, que se muestran en la consola del sistema y en el archivo `/var/adm/messages`.

Esta sección describe cómo determinar el estado de agrupaciones y dispositivos. En este capítulo no se explica cómo reparar o recuperarse de agrupaciones cuyo estado es defectuoso. Si desea más información sobre cómo resolver problemas y recuperar datos, consulte el Capítulo 11, “Recuperación de agrupaciones y solución de problemas de Oracle Solaris ZFS”.

Cada dispositivo puede tener uno de los estados siguientes:

`ONLINE` El dispositivo o dispositivo virtual funciona normalmente. Quizá haya algunos errores transitorios, pero el dispositivo funciona.

`DEGRADED` El dispositivo virtual ha sufrido un fallo pero sigue funcionando. Es el estado más habitual si un dispositivo RAID-Z o una duplicación pierden uno o más dispositivos constituyentes. La tolerancia a errores de la agrupación puede verse comprometida: un error posterior en otro dispositivo puede llegar a ser irrecuperable.

FAULTED	No se puede acceder al dispositivo o dispositivo virtual. Este estado suele denotar un error total del dispositivo, por ejemplo ZFS es incapaz de enviar o recibir datos del dispositivo. Si un dispositivo virtual de nivel superior se encuentra en este estado, no hay forma de acceder a la agrupación.
OFFLINE	El administrador ha dejado expresamente sin conexión el dispositivo
UNAVAIL	El dispositivo o dispositivo virtual no se puede abrir. En algunos casos, las agrupaciones con dispositivos en estado UNAVAIL se muestran en modo DEGRADED. Si un dispositivo virtual de nivel superior tiene estado UNAVAIL, la agrupación queda completamente inaccesible.
REMOVED	Se ha extraído físicamente el dispositivo mientras el sistema estaba ejecutándose. La detección de extracción de dispositivos depende del hardware y quizá no se admita en todas las plataformas.

El estado de una agrupación lo determina el estado de todos sus dispositivos virtuales de nivel superior. Si todos los dispositivos virtuales están ONLINE, la agrupación también está ONLINE. Si uno de los dispositivos virtuales tiene el estado DEGRADED o UNAVAIL, la agrupación también tiene el estado DEGRADED. Si un dispositivo virtual de nivel superior tiene el estado FAULTED u OFFLINE, la agrupación también tiene el estado FAULTED. Una agrupación con estado FAULTED es completamente inaccesible. La recuperación de datos no es factible hasta que los dispositivos necesarios se conectan o reparan. Una agrupación con estado DEGRADED sigue funcionando, pero quizá no obtenga el mismo nivel de redundancia o rendimiento de datos que si tuviera conexión.

Estado de la agrupación de almacenamiento básico

El modo más rápido de averiguar el estado de salud de agrupaciones consiste en usar el comando `zpool status` como se indica a continuación:

```
# zpool status -x
all pools are healthy
```

Si desea examinar una determinada agrupación, indique su nombre en la sintaxis de comando. Cualquier agrupación que no esté en estado ONLINE debe comprobarse para descartar problemas potenciales, tal como se explica en la sección siguiente.

Estado detallado

Puede solicitar un resumen de estado más detallado mediante la opción `-v`. Por ejemplo:

```
# zpool status -v tank
  pool: tank
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
```

```
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: scrub completed after 0h0m with 0 errors on Wed Jan 20 15:13:59 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t1d0  UNAVAIL      0     0     0  cannot open

errors: No known data errors
```

Esta salida muestra la descripción completa de por qué la agrupación se encuentra en un estado determinado, incluida una descripción legible del problema y un vínculo a un artículo sobre la materia para obtener más información. Cada artículo técnico ofrece información actualizada sobre el mejor método de resolución del problema actual. El uso de la información de configuración detallada permite determinar el dispositivo dañado y la forma de reparar la agrupación.

En el ejemplo anterior, el dispositivo defectuoso se debe sustituir. Una vez reemplazado, utilice el comando `zpool online` para que el dispositivo se conecte de nuevo. Por ejemplo:

```
# zpool online tank c1t0d0
Bringing device c1t0d0 online
# zpool status -x
all pools are healthy
```

Si la propiedad `autoreplace` está activada, es posible que no sea necesario conectar el dispositivo reemplazado.

Si una agrupación tiene un dispositivo sin conexión, la salida del comando identifica la agrupación problemática. Por ejemplo:

```
# zpool status -x
  pool: tank
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
 scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 15:15:09 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t1d0  OFFLINE      0     0     0  48K resilvered

errors: No known data errors
```

Las columnas `READ` y `WRITE` ofrecen un recuento de errores de E/S producidos en el dispositivo; y la columna `CKSUM` ofrece un recuento de errores de suma de comprobación del dispositivo que no pueden corregirse. Ambos recuentos de errores indican un error potencial del dispositivo y las pertinentes acciones correctivas. Si se informa de que un dispositivo virtual de nivel superior tiene de errores distintos de cero, quizá ya no se pueda acceder a algunas porciones de datos.

El campo `errors:` identifica cualquier error de datos conocido.

En la salida del ejemplo anterior, el dispositivo que no está conectado no provoca errores de datos.

Para obtener más información sobre diagnósticos y reparaciones de datos y agrupaciones defectuosos, consulte el Capítulo 11, "Recuperación de agrupaciones y solución de problemas de Oracle Solaris ZFS".

Migración de agrupaciones de almacenamiento de ZFS

En ocasiones puede ser preciso mover una agrupación de almacenamiento de un sistema a otro. Para hacerlo, los dispositivos de almacenamiento se deben desconectar del sistema original y volver a conectar en el de destino. Esta tarea se debe efectuar mediante el recableado físico de los dispositivos o mediante dispositivos con varios puertos como los de una SAN. ZFS permite exportar la agrupación de un sistema e importarla al de destino, incluso si los sistemas tienen un orden diferente de almacenamiento de una secuencia de datos en la memoria. Si desea información sobre cómo repetir o migrar sistemas de archivos entre distintas agrupaciones de almacenamiento que podrían estar en equipos distintos, consulte "Envío y recepción de datos ZFS" en la página 234.

Preparación para la migración de agrupaciones de almacenamiento de ZFS

Las agrupaciones de almacenamiento se deben exportar para indicar que están preparadas para la migración. Esta operación purga cualquier dato no escrito en el disco, escribe datos en el disco para indicar que la exportación se ha realizado y elimina del sistema cualquier información de la agrupación.

Si no exporta la agrupación, sino que elimina manualmente los discos, aún es posible importar la agrupación resultante en otro sistema. Sin embargo, podría perder los últimos segundos de transacciones de datos, con lo cual la agrupación aparece como defectuosa en el sistema original debido a que los dispositivos ya no están presentes. De forma predeterminada, el sistema de destino es incapaz de importar una agrupación que no se ha exportado explícitamente. Esta condición es necesaria para impedir la importación accidental de una agrupación activa con almacenamiento conectado a la red que todavía se utilice en otro sistema.

Exportación de una agrupación de almacenamiento de ZFS

Si desea exportar una agrupación, utilice el comando `zpool export`. Por ejemplo:

```
# zpool export tank
```

Antes de continuar, el comando intenta desmontar cualquier sistema de archivos montado en la agrupación. Si alguno de los sistemas de archivos no consigue desmontarse, puede forzar el desmontaje mediante la opción `-f`. Por ejemplo:

```
# zpool export tank
cannot unmount '/export/home/eschrock': Device busy
# zpool export -f tank
```

Tras ejecutar este comando, la agrupación `tank` deja de estar visible en el sistema.

Si al exportar hay dispositivos no disponibles, no se pueden especificar como exportados correctamente. Si uno de estos dispositivos se conecta más adelante a un sistema sin uno de los dispositivos en funcionamiento, aparece como "potencialmente activo".

Si los volúmenes de ZFS se utilizan en la agrupación, ésta no se puede exportar, ni siquiera con la opción `-f`. Para exportar una agrupación con un volumen de ZFS, antes debe comprobar que no esté activo ninguno de los consumidores del volumen.

Para obtener más información sobre los volúmenes de ZFS, consulte "Volúmenes de ZFS" en la página 281.

Especificación de agrupaciones de almacenamiento disponibles para importar

Cuando la agrupación se haya eliminado del sistema (ya sea al exportar explícitamente o eliminar dispositivos de manera forzada), conecte los dispositivos al sistema de destino. ZFS puede controlar determinadas situaciones en que sólo algunos de los dispositivos están disponibles, pero una migración de agrupaciones correcta depende de la salud global de los dispositivos. Además, no es esencial que los dispositivos estén vinculados bajo el mismo nombre de dispositivo. ZFS detecta cualquier dispositivo que se haya movido o al que se haya cambiado el nombre, y ajusta la configuración en consonancia. Para detectar las agrupaciones disponibles, ejecute el comando `zpool import` sin opciones. Por ejemplo:

```
# zpool import
 pool: tank
    id: 11809215114195894163
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        tank        ONLINE
          mirror-0  ONLINE
            c1t0d0  ONLINE
            c1t1d0  ONLINE
```

En este ejemplo, la agrupación `tank` está disponible para importarla al sistema de destino. Cada agrupación está identificada mediante un nombre, así como un identificador numérico exclusivo. Si hay varias agrupaciones para importar con el mismo nombre, puede utilizar el identificador numérico para diferenciarlas.

De forma parecida a la salida del comando `zpool status`, la salida `zpool import` incluye un vínculo a un artículo divulgativo con la información más actualizada sobre procedimientos de resolución de un problema que impide la importación de una agrupación. En este caso, el usuario puede forzar la importación de una agrupación. Sin embargo, importar una agrupación que utiliza otro sistema en una red de almacenamiento puede dañar datos y generar avisos graves del sistema, puesto que ambos sistemas intentan escribir en el mismo almacenamiento. Si algunos dispositivos de la agrupación no están disponibles pero hay suficiente redundancia para tener una agrupación utilizable, la agrupación mostrará el estado `DEGRADED`. Por ejemplo:

```
# zpool import
  pool: tank
    id: 11809215114195894163
 state: DEGRADED
status: One or more devices are missing from the system.
action: The pool can be imported despite missing or damaged devices.  The
        fault tolerance of the pool may be compromised if imported.
   see: http://www.sun.com/msg/ZFS-8000-2Q
config:

        NAME        STATE     READ WRITE CKSUM
```

```
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c1t0d0  UNAVAIL      0     0     0  cannot open
            c1t3d0  ONLINE       0     0     0
```

En este ejemplo, el primer disco está dañado o no se encuentra, aunque aún puede importar la agrupación porque todavía se puede acceder a los datos reflejados. Si faltan muchos dispositivos o hay demasiados defectuosos, la agrupación no se puede importar. Por ejemplo:

```
# zpool import
  pool: dozer
    id: 9784486589352144634
 state: FAULTED
action: The pool cannot be imported. Attach the missing
        devices and try again.
   see: http://www.sun.com/msg/ZFS-8000-6X
config:
        raidz1-0       FAULTED
          c1t0d0       ONLINE
          c1t1d0       FAULTED
          c1t2d0       ONLINE
          c1t3d0       FAULTED
```

En este ejemplo faltan dos discos de un dispositivo virtual RAID-Z. Eso significa que no hay suficientes datos redundantes disponibles para reconstruir la agrupación. En algunos casos no hay suficientes dispositivos para determinar la configuración completa. En este caso, ZFS desconoce los demás dispositivos que formaban parte de la agrupación, aunque ZFS proporciona todos los datos posibles relativos a la situación. Por ejemplo:

```
# zpool import
pool: dozer
    id: 9784486589352144634
 state: FAULTED
status: One or more devices are missing from the system.
action: The pool cannot be imported. Attach the missing
        devices and try again.
   see: http://www.sun.com/msg/ZFS-8000-6X
config:
        dozer          FAULTED   missing device
          raidz1-0     ONLINE
            c1t0d0     ONLINE
            c1t1d0     ONLINE
            c1t2d0     ONLINE
            c1t3d0     ONLINE
Additional devices are known to be part of this pool, though their
exact configuration cannot be determined.
```

Importación de agrupaciones de almacenamiento de ZFS de directorios alternativos

De modo predeterminado, el comando `zpool import` sólo busca dispositivos en el directorio `/dev/dsk`. Si los dispositivos existen en otro directorio, o si utiliza agrupaciones de las que se ha hecho copia de seguridad mediante archivos, utilice la opción `-d` para buscar en directorios alternativos. Por ejemplo:

```
# zpool create dozer mirror /file/a /file/b
# zpool export dozer
# zpool import -d /file
  pool: dozer
    id: 7318163511366751416
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        dozer        ONLINE
          mirror-0   ONLINE
            /file/a  ONLINE
            /file/b  ONLINE
# zpool import -d /file dozer
```

Si los dispositivos están en varios directorios, puede especificar múltiples opciones de `-d`.

Importación de agrupaciones de almacenamiento de ZFS

Tras identificar una agrupación para importarla, debe especificar el nombre de la agrupación o su identificador numérico como argumento en el comando `zpool import`. Por ejemplo:

```
# zpool import tank
```

Si hay varias agrupaciones con el mismo nombre, indique la agrupación que desea importar mediante el identificador numérico. Por ejemplo:

```
# zpool import
  pool: dozer
    id: 2704475622193776801
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        dozer       ONLINE
          c1t9d0    ONLINE

  pool: dozer
    id: 6223921996155991199
```

```
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        dozer       ONLINE
          c1t8d0    ONLINE
# zpool import dozer
cannot import 'dozer': more than one matching pool
import by numeric ID instead
# zpool import 6223921996155991199
```

Si el nombre de la agrupación entra en conflicto con un nombre de agrupación que ya existe, puede importarlo con otro nombre. Por ejemplo:

```
# zpool import dozer zeepool
```

Este comando importa la agrupación exportada `dozer` con el nombre nuevo `zeepool`.

Si la agrupación no se ha exportado correctamente, ZFS solicita que el indicador `-f` impida la importación accidental de una agrupación que otro sistema todavía está usando. Por ejemplo:

```
# zpool import dozer
cannot import 'dozer': pool may be in use on another system
use '-f' to import anyway
# zpool import -f dozer
```

Las agrupaciones también se pueden importar en una raíz alternativa mediante la opción `-R`. Si desea más información sobre otras agrupaciones raíz, consulte "Uso de agrupaciones raíz de ZFS alternativas" en la página 289.

Recuperación de agrupaciones de almacenamiento de ZFS destruidas

El comando `zpool import -D` es apto para recuperar una agrupación de almacenamiento que se haya destruido. Por ejemplo:

```
# zpool destroy tank
# zpool import -D
  pool: tank
    id: 5154272182900538157
 state: ONLINE (DESTROYED)
action: The pool can be imported using its name or numeric identifier.
config:

        tank        ONLINE
          mirror-0  ONLINE
            c1t0d0  ONLINE
            c1t1d0  ONLINE
```

En esta salida `zpool import`, puede identificar la agrupación `tank` como la destruida debido a la siguiente información de estado:

```
state: ONLINE (DESTROYED)
```

Para recuperar la agrupación destruida, ejecute de nuevo el comando `zpool import -D` con la agrupación que se debe recuperar. Por ejemplo:

```
# zpool import -D tank
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE
          mirror-0  ONLINE
            c1t0d0  ONLINE
            c1t1d0  ONLINE

errors: No known data errors
```

La agrupación destruida se puede recuperar aunque uno de los dispositivos de esta agrupación sea defectuoso o no esté disponible, mediante la inclusión de la opción `-f`. En esta situación, debería importar la agrupación degradada y después intentar solucionar el error de dispositivo. Por ejemplo:

```
# zpool destroy dozer
# zpool import -D
pool: dozer
    id: 13643595538644303788
 state: DEGRADED (DESTROYED)
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
config:

        NAME         STATE     READ WRITE CKSUM
        dozer        DEGRADED     0     0     0
          raidz2-0   DEGRADED     0     0     0
            c2t8d0   ONLINE       0     0     0
            c2t9d0   ONLINE       0     0     0
            c2t10d0  ONLINE       0     0     0
            c2t11d0  UNAVAIL      0    35     1  cannot open
            c2t12d0  ONLINE       0     0     0

errors: No known data errors
# zpool import -Df dozer
# zpool status -x
  pool: dozer
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
```

```
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: scrub completed after 0h0m with 0 errors on Thu Jan 21 15:38:48 2010
config:

        NAME         STATE     READ WRITE CKSUM
        dozer        DEGRADED     0     0     0
          raidz2-0   DEGRADED     0     0     0
            c2t8d0   ONLINE       0     0     0
            c2t9d0   ONLINE       0     0     0
            c2t10d0  ONLINE       0     0     0
            c2t11d0  UNAVAIL      0    37     0  cannot open
            c2t12d0  ONLINE       0     0     0

errors: No known data errors
# zpool online dozer c2t11d0
Bringing device c2t11d0 online
# zpool status -x
all pools are healthy
```

Actualización de agrupaciones de almacenamiento de ZFS

Si dispone de agrupaciones de almacenamiento de ZFS de una versión anterior (por ejemplo, Solaris 10 10/09) las agrupaciones pueden actualizarse con el comando `zpool upgrade` para poder aprovechar las funciones de las agrupaciones de la versión actual Asimismo, el comando `zpool status` se ha modificado para notificar a los usuarios que las agrupaciones están ejecutando versiones antiguas. Por ejemplo:

```
# zpool status
  pool: tank
 state: ONLINE
status: The pool is formatted using an older on-disk format.  The pool can
        still be used, but some features are unavailable.
action: Upgrade the pool using 'zpool upgrade'.  Once this is done, the
        pool will no longer be accessible on older software versions.
 scrub: none requested
config:
        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
errors: No known data errors
```

La sintaxis siguiente es válida para identificar información adicional sobre una versión concreta y compatible:

```
# zpool upgrade -v
This system is currently running ZFS pool version 22.

The following versions are supported:
```

```
VER  DESCRIPTION
---  --------------------------------------------------------
 1   Initial ZFS version
 2   Ditto blocks (replicated metadata)
 3   Hot spares and double parity RAID-Z
 4   zpool history
 5   Compression using the gzip algorithm
 6   bootfs pool property
 7   Separate intent log devices
 8   Delegated administration
 9   refquota and refreservation properties
 10  Cache devices
 11  Improved scrub performance
 12  Snapshot properties
 13  snapused property
 14  passthrough-x aclinherit
 15  user/group space accounting
 16  stmf property support
 17  Triple-parity RAID-Z
 18  Snapshot user holds
 19  Log device removal
 20  Compression using zle (zero-length encoding)
 21  Reserved
 22  Received properties

For more information on a particular version, including supported releases,
see the ZFS Administration Guide.
```

A continuación puede ejecutar el comando `zpool upgrade` para actualizar todas las agrupaciones. Por ejemplo:

```
# zpool upgrade -a
```

Nota – Si moderniza la agrupación a una versión de ZFS posterior, no se podrá acceder a la agrupación en un sistema que ejecute una versión antigua de ZFS.

◆ ◆ ◆ CAPÍTULO 5

Instalación e inicio de un sistema de archivos raíz ZFS Oracle Solaris

En este capítulo se describe cómo instalar e iniciar un sistema de archivos ZFS Oracle Solaris. También se describe la migración de un sistema de archivos raíz UFS a un sistema de archivos ZFS mediante Actualización automática de Oracle Solaris.

Este capítulo se divide en las secciones siguientes:

Si desea obtener una lista de problemas conocidos de esta versión, consulte *Notas de la versión de Oracle Solaris 10 9/10*.

Para obtener información actualizada sobre resolución de problemas, consulte el sitio siguiente:

`http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide`

Instalación e inicio de un sistema de archivos raíz ZFS Oracle Solaris (información general)

A partir de Solaris 10 10/08, se puede llevar a cabo una instalación e iniciar desde un sistema de archivos raíz ZFS de las maneras siguientes:

- Mediante una instalación inicial durante la que se selecciona ZFS como sistema de archivos raíz.
- Mediante Actualización automática de Oracle Solaris para migrar de un sistema de archivos raíz UFS a uno ZFS. Además, Actualización automática de Oracle Solaris es apta para efectuar las tareas siguientes:
  - Crear un entorno de inicio a partir de una agrupación raíz ZFS ya existente.
  - Crear un entorno de inicio en una agrupación raíz ZFS nueva.
- Puede utilizar un perfil de Oracle Solaris JumpStart para instalar automáticamente un sistema con un sistema de archivos raíz ZFS.
- A partir de la versión Solaris 10 10/09, puede utilizar un perfil JumpStart para instalar automáticamente un sistema con un archivo de almacenamiento flash ZFS.

Después de instalar un sistema basado en SPARC o x86 con un sistema de archivos raíz ZFS o de migrar a un sistema de archivos raíz ZFS, el sistema se inicia automáticamente desde el sistema de archivos raíz ZFS. Para obtener más información sobre cambios de inicio, consulte "Inicio desde un sistema de archivos raíz ZFS" en la página 170.

Funciones de instalación de ZFS

En esta versión de Solaris se proporcionan las siguientes funciones de instalación de ZFS:

- El instalador de texto interactivo de Solaris permite instalar un sistema de archivos raíz UFS o ZFS. En esta versión de Solaris, UFS sigue siendo el sistema de archivos predeterminado. Hay varias formas de acceder a la opción del instalador de texto interactivo:
  - SPARC: utilice la sintaxis siguiente desde el DVD de instalación de Solaris:

    ```
    ok boot cdrom - text
    ```
  - SPARC: utilice la sintaxis siguiente cuando inicie desde la red:

    ```
    ok boot net - text
    ```
  - x86: seleccione la opción de instalación en modo texto.
- Un perfil JumpStart personalizado proporciona las siguientes funciones:
  - Puede configurar un perfil para crear una agrupación de almacenamiento ZFS y designar un sistema de archivos ZFS de inicio.
  - Se puede configurar un perfil con el fin de identificar un archivo de almacenamiento flash de una agrupación raíz ZFS.

- Con Automatización automática de Oracle Solaris se puede migrar de un sistema de archivos raíz UFS a uno ZFS. Los comandos `lucreate` y `luactivate` se han mejorado para admitir sistemas de archivos y agrupaciones ZFS.
- Se puede configurar una agrupación raíz ZFS reflejada seleccionando dos discos durante la instalación. También se pueden vincular más discos después de la instalación para crear una agrupación raíz ZFS reflejada.
- Los dispositivos de volcado e intercambio se crean de manera automática en volúmenes ZFS de la agrupación raíz ZFS:

En esta versión no se proporcionan las siguientes funciones de instalación:

- No está disponible la función de instalación de GUI para instalar un sistema de archivos raíz ZFS.
- La función de instalación flash de Oracle Solaris para instalar un sistema de archivos raíz ZFS no está disponible mediante la selección de la opción de instalación flash de la opción de instalación inicial. Sin embargo, puede crear un perfil JumpStart para identificar un archivo de almacenamiento flash de una agrupación raíz ZFS. Para obtener más información, consulte "Instalación de un sistema de archivos raíz ZFS (instalación de archivo de almacenamiento flash de Oracle Solaris)" en la página 137.
- El programa de actualización estándar no es válido para actualizar el sistema de archivos raíz UFS a un sistema de archivos raíz ZFS.

Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS

Antes de intentar instalar un sistema con un sistema de archivos raíz ZFS o de migrar un sistema de archivos raíz UFS a uno ZFS, deben cumplirse los requisitos siguientes:

Requisitos de la versión de Oracle Solaris

Puede instalar e iniciar un sistema de archivos raíz ZFS, o bien migrar a un sistema de archivos raíz ZFS de las maneras siguientes:

- Instalación de un sistema de archivos raíz ZFS: disponible a partir de la versión Solaris 10 10/08.
- Migración de un sistema de archivos raíz UFS a un sistema de archivos raíz ZFS mediante Automatización automática de Oracle Solaris: debe tener instalado al menos Solaris 10 10/08, o bien debe haber actualizado Solaris al menos a la versión 10 10/08.

Requisitos generales de la agrupación de almacenamiento ZFS

Las siguientes secciones describen los requisitos de configuración y el espacio de la agrupación raíz ZFS.

Requisitos de espacio en el disco para agrupaciones de almacenamiento ZFS

La cantidad mínima necesaria de espacio de agrupación para un sistema de archivos raíz ZFS es mayor que la de un sistema de archivos raíz UFS porque los dispositivos de intercambio y volcado deben ser independientes en un entorno raíz ZFS. De forma predeterminada, en un sistema de archivos raíz UFS los dispositivos de intercambio y volcado son el mismo dispositivo.

Al instalar o actualizar un sistema con un sistema de archivos raíz ZFS, el tamaño del área de intercambio y del dispositivo de volcado dependen de la cantidad de memoria física. La cantidad mínima de espacio de agrupación disponible para un sistema de archivos raíz ZFS depende de la cantidad de memoria física, el espacio en el disco disponible y la cantidad de entornos de inicio que se vayan a crear.

Revise los siguientes requisitos de espacio en el disco para agrupaciones de almacenamiento ZFS:

- Para instalar un sistema de archivos raíz ZFS se necesita como mínimo 768 MB.
- Se recomienda 1 GB de memoria para mejorar el rendimiento global de ZFS.
- Se recomienda un mínimo de 16 GB de espacio en el disco. El espacio en el disco se consume del modo siguiente:
  - **Área de intercambio y dispositivo de volcado**: los tamaños predeterminados de los volúmenes de intercambio y volcado que se crean mediante el programa de instalación de Solaris son los siguientes:
    - **Instalación inicial de Solaris**, en el nuevo entorno de inicio ZFS, el tamaño del volumen de intercambio predeterminado se calcula como la mitad del tamaño de la memoria física, normalmente en el intervalo de 512 MB a 2 GB. Durante una instalación inicial se puede ajustar el tamaño de intercambio.
    - El tamaño predeterminado del volumen de volcado se calcula mediante el núcleo, en función de la información de `dumpadm` y el tamaño de la memoria física. Durante una instalación inicial se puede ajustar el tamaño de volcado.
    - **Actualización automática de Oracle Solaris**: si un sistema de archivos raíz UFS se migra a un sistema de archivos raíz ZFS, el tamaño predeterminado del volumen de intercambio del entorno de inicio ZFS se calcula como el tamaño del dispositivo de intercambio del entorno de inicio UFS. El cálculo del tamaño predeterminado del volumen de intercambio suma los tamaños de todos los dispositivos de intercambio del entorno de inicio UFS y crea un volumen ZFS de ese tamaño en el entorno de inicio ZFS. Si en el entorno de inicio UFS no se definen dispositivos de intercambio, el tamaño predeterminado del volumen de intercambio se establece en 512 MB.

- En el entorno de inicio ZFS, el tamaño del volumen de volcado predeterminado se establece en la mitad del tamaño de la memoria física, entre 512 MB y 2 GB.

Puede ajustar los tamaños de los volúmenes de intercambio y volcado según lo que necesite, siempre y cuando los nuevos tamaños permitan el funcionamiento del sistema. Para obtener más información, consulte "Ajuste del tamaño de los dispositivos de intercambio y volcado ZFS" en la página 168.

- **Entorno de inicio**: aparte de nuevos requisitos de espacio de intercambio o volcado, o de tamaños de dispositivos de intercambio o volcado, un entorno de inicio ZFS que se migra de UFS necesita unos 6 GB. Cada entorno de inicio ZFS que se clona de otro entorno de inicio ZFS no necesita espacio en el disco adicional, pero se debe tener en cuenta que el tamaño del entorno de inicio aumentará al aplicarse parches. Todos los entornos de inicio ZFS de la misma agrupación raíz deben utilizar los mismos dispositivos de intercambio y volcado.
- **Componentes del sistema operativo Solaris**: todos los subdirectorios del sistema de archivos raíz que forman parte de la imagen del sistema operativo, con la excepción de `/var`, deben estar en el mismo conjunto de datos que el sistema de archivos raíz. Además, todos los componentes del sistema operativo Solaris deben residir en la agrupación raíz, con la excepción de los dispositivos de intercambio y volcado.

  Otra restricción es que el directorio o el conjunto de datos`/var` debe ser un único conjunto de datos. Por ejemplo, no puede crear un conjunto de datos `/var` descendiente, como, por ejemplo, `/var/tmp`, si desea utilizar también Actualización automática de Oracle Solaris para migrar o parchear un entorno de inicio ZFS o crear un archivo de almacenamiento flash ZFS de esta agrupación.

Por ejemplo, un sistema con 12 GB de espacio en el disco puede ser demasiado pequeño para un entorno ZFS que se puede iniciar, ya que se necesitan 2 GB de espacio en el disco para cada dispositivo de intercambio y volcado, así como unos 6 GB de espacio en el disco para el entorno de inicio ZFS que se migra de un entorno de inicio UFS.

Requisitos de configuración de la agrupación de almacenamiento ZFS

Revise los siguientes requisitos de configuración de la agrupación de almacenamiento ZFS:

- La agrupación que está destinada a ser la agrupación raíz debe tener una etiqueta SMI. Este requisito se cumple si la agrupación se crea con segmentos de disco.
- La agrupación debe existir ya sea en un segmento de disco o en segmentos de disco que se han reflejado. Si en el transcurso de una migración con Actualización automática de Oracle Solaris se intenta utilizar una configuración de agrupación no permitida, aparecerá un mensaje similar al siguiente:

  ```
  ERROR: ZFS pool name does not support boot environments
  ```

  Para obtener una descripción detallada de las configuraciones admitidas para la agrupación raíz ZFS, consulte "Creación de una agrupación raíz ZFS" en la página 74.

- x86: el disco debe contener una partición `fdisk` de Solaris. Se crea una partición `fdisk` de Solaris automáticamente cuando se instala el sistema basado en x86. Para obtener más información acerca de las particiones `fdisk` de Solaris, consulte "Guidelines for Creating an fdisk Partition" de *System Administration Guide: Devices and File Systems*.
- Los discos destinados a iniciarse en una agrupación raíz ZFS deben tener como límite de tamaño 1 TB, tanto en sistemas SPARC como x86.
- La compresión puede habilitarse en la agrupación raíz, pero sólo después de que se haya instalado la agrupación raíz. No hay forma de habilitar la compresión en una agrupación raíz durante la instalación. El algoritmo de compresión `gzip` no se admite en las agrupaciones raíz.
- No cambie el nombre de la agrupación raíz después de su creación mediante una instalación inicial o tras la migración con Actualización automática de Oracle Solaris a un sistema de archivos raíz ZFS. El cambio de la agrupación raíz puede impedir el inicio del sistema.

Instalación de un sistema de archivos raíz ZFS (instalación inicial)

En esta versión de Solaris, puede efectuar una instalación inicial mediante el instalador de texto interactivo de Solaris para crear una agrupación de almacenamiento ZFS que contenga un sistema de archivos raíz ZFS que se pueda iniciar. Si dispone de una agrupación de almacenamiento ZFS que desea utilizar en el sistema de archivos raíz ZFS, debe emplear Actualización automática de Oracle Solaris para migrar del sistema de archivos raíz UFS actual a uno ZFS que haya en una agrupación de almacenamiento ZFS. Para obtener más información, consulte "Migración de un sistema de archivos raíz UFS a uno ZFS (Actualización automática de Oracle Solaris)" en la página 144.

Si va a configurar las zonas después de la instalación inicial de un sistema de archivos raíz ZFS y tiene previsto aplicar parches o actualizaciones al sistema, consulte "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)" en la página 151 o "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156.

Si ya tiene agrupaciones de almacenamiento ZFS en el sistema, se confirman con el siguiente mensaje. Sin embargo, estas agrupaciones permanecen intactas, a menos que se seleccionen los discos de las agrupaciones existentes para crear la nueva agrupación de almacenamiento.

```
There are existing ZFS pools available on this system.  However, they can only be upgraded
using the Live Upgrade tools.  The following screens will only allow you to install a ZFS root system,
not upgrade one.
```

Precaución – Las agrupaciones que existan se destruirán si para la nueva agrupación se selecciona cualquiera de sus discos.

Antes de comenzar la instalación inicial para crear una agrupación de almacenamiento ZFS, consulte "Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS" en la página 127.

EJEMPLO 5–1 Instalación inicial de un sistema de archivos raíz ZFS que se puede iniciar

El proceso de instalación de texto interactivo de Solaris es, básicamente, el mismo que el de anteriores versiones de Solaris, salvo el hecho de indicar al usuario que cree un sistema de archivos raíz UFS o ZFS. En esta versión, UFS sigue siendo el sistema de archivos predeterminado. Si se selecciona un sistema de archivos raíz ZFS, se indica al usuario que cree una agrupación de almacenamiento ZFS. Los pasos necesarios para instalar un sistema de archivos raíz ZFS se indican a continuación:

1. Seleccione el método de instalación interactiva de Solaris porque para crear un sistema de archivos raíz ZFS que se puede iniciar no se dispone de la instalación flash de Solaris. Sin embargo, puede crear un archivo de almacenamiento flash ZFS para utilizarlo durante una instalación JumpStart. Para obtener más información, consulte "Instalación de un sistema de archivos raíz ZFS (instalación de archivo de almacenamiento flash de Oracle Solaris)" en la página 137.

   A partir de la versión Solaris 10 10/08, puede migrar un sistema de archivos raíz UFS a un sistema de archivos raíz ZFS siempre que ya se haya instalado al menos la versión Solaris 10 10/08. Para obtener más información sobre cómo migrar a un sistema de archivos raíz ZFS, consulte "Migración de un sistema de archivos raíz UFS a uno ZFS (Actualización automática de Oracle Solaris)" en la página 144.

2. Para crear un sistema de archivos raíz ZFS, seleccione la opción ZFS. Por ejemplo:

```
Choose Filesystem Type

  Select the filesystem to use for your Solaris installation


            [ ] UFS
            [X] ZFS
```

3. Una vez seleccionado el software que se instalará, se le pedirá que seleccione los discos para crear la agrupación de almacenamiento ZFS. Esta pantalla es similar a la de las versiones anteriores de Solaris.

```
Select Disks

  On this screen you must select the disks for installing Solaris software.
  Start by looking at the Suggested Minimum field; this value is the
  approximate space needed to install the software you've selected. For ZFS,
  multiple disks will be configured as mirrors, so the disk you choose, or the
  slice within the disk must exceed the Suggested Minimum value.
```

EJEMPLO 5–1 Instalación inicial de un sistema de archivos raíz ZFS que se puede iniciar *(Continuación)*

```
NOTE: ** denotes current boot disk

Disk Device                                                 Available Space
=============================================================================
[X]    c1t0d0                                              69994 MB  (F4 to edit)
[ ]    c1t1d0                                              69994 MB
[-]    c1t2d0                                                  0 MB
[-]    c1t3d0                                                  0 MB

                                  Maximum Root Size:  69994 MB
                                  Suggested Minimum:   8279 MB
```

Puede seleccionar el disco o los discos que deben usarse para la agrupación raíz ZFS. Si selecciona dos discos, para la agrupación raíz se establece una configuración de dos discos reflejados. La configuración óptima es una agrupación de dos o tres discos reflejados. Si tiene ocho discos y los selecciona todos, se utilizan para la agrupación raíz como un gran reflejo. Esta configuración no es óptima. Otra opción es crear una agrupación raíz reflejada cuando se haya terminado la instalación inicial. No es posible efectuar una configuración de agrupaciones RAID-Z para la agrupación raíz. Si desea más información sobre la configuración de agrupaciones de almacenamiento ZFS, consulte "Funciones de repetición de una agrupación de almacenamiento de ZFS" en la página 69.

4. Para seleccionar dos discos para crear una agrupación raíz reflejada, utilice las teclas de control del cursor para seleccionar el segundo disco. En el ejemplo siguiente, tanto `c1t1d0` como `c1t2d0` se seleccionan como los discos de la agrupación raíz. Los dos discos deben tener una etiqueta SMI y un segmento 0. Si los discos no están etiquetados con una etiqueta SMI o no contienen segmentos, debe salir del programa de instalación, usar la utilidad `format` para reetiquetar y reparticionar los discos y, a continuación, reiniciar el programa de instalación.

```
Select Disks

  On this screen you must select the disks for installing Solaris software.
  Start by looking at the Suggested Minimum field; this value is the
  approximate space needed to install the software you've selected. For ZFS,
  multiple disks will be configured as mirrors, so the disk you choose, or the
  slice within the disk must exceed the Suggested Minimum value.
  NOTE: ** denotes current boot disk

  Disk Device                                                 Available Space
  =============================================================================
  [X]    c1t0d0                                              69994 MB
  [X]    c1t1d0                                              69994 MB  (F4 to edit)
  [-]    c1t2d0                                                  0 MB
  [-]    c1t3d0                                                  0 MB

                                    Maximum Root Size:  69994 MB
                                    Suggested Minimum:   8279 MB
```

Si la columna Esp. disponible identifica 0 MB, es muy probable que el disco tenga una etiqueta EFI. Si desea utilizar un disco con una etiqueta EFI, deberá salir del programa de

EJEMPLO 5–1 Instalación inicial de un sistema de archivos raíz ZFS que se puede iniciar *(Continuación)*

instalación, volver a etiquetar el disco con una etiqueta SMI utilizando el comando `format -e` y, a continuación, reiniciar el programa de instalación.

Si no crea una agrupación raíz reflejada durante la instalación, puede crear fácilmente una después de la instalación. Para obtener información, consulte "Cómo crear una agrupación raíz reflejada (posterior a la instalación)" en la página 136.

5. Tras haber seleccionado uno o varios discos para la agrupación de almacenamiento ZFS, aparece una pantalla similar a la siguiente:

```
Configure ZFS Settings

  Specify the name of the pool to be created from the disk(s) you have chosen.
  Also specify the name of the dataset to be created within the pool that is
  to be used as the root directory for the filesystem.

              ZFS Pool Name: rpool
      ZFS Root Dataset Name: s10s_u9wos_08
      ZFS Pool Size (in MB): 69995
  Size of Swap Area (in MB): 2048
  Size of Dump Area (in MB): 1536
        (Pool size must be between 6231 MB and 69995 MB)

                        [X] Keep / and /var combined
                        [ ] Put /var on a separate dataset
```

En esta pantalla se puede cambiar el nombre de la agrupación ZFS, el nombre del conjunto de datos, el tamaño de la agrupación y el tamaño de los dispositivos de intercambio y volcado. Para ello, con las teclas de control del cursor desplácese por las entradas y sustituya los valores predeterminados por los nuevos. Si lo desea, puede aceptar los valores predeterminados. Además, puede modificar el modo de crear y montar el sistema de archivos `/var`.

En este ejemplo, el nombre del conjunto de datos raíz se cambia a `zfsBE`.

```
              ZFS Pool Name: rpool
      ZFS Root Dataset Name: zfsBE
      ZFS Pool Size (in MB): 69995
  Size of Swap Area (in MB): 2048
  Size of Dump Area (in MB): 1536
        (Pool size must be between 6231 MB and 69995 MB)

                        [X] Keep / and /var combined
                        [ ] Put /var on a separate dataset
```

6. El perfil de instalación se puede cambiar en esta pantalla final de la instalación. Por ejemplo:

```
Profile

  The information shown below is your profile for installing Solaris software.
  It reflects the choices you've made on previous screens.

  ============================================================================
```

EJEMPLO 5–1 Instalación inicial de un sistema de archivos raíz ZFS que se puede iniciar *(Continuación)*

```
                  Installation Option: Initial
                          Boot Device: c1t0d0
                Root File System Type: ZFS
                      Client Services: None

                              Regions: North America
                        System Locale: C ( C )

                             Software: Solaris 10, Entire Distribution
                            Pool Name: rpool
                Boot Environment Name: zfsBE
                            Pool Size: 69995 MB
                      Devices in Pool: c1t0d0
                                       c1t1d0
```

7. Una vez finalizada la instalación, examine la información del sistema de archivos y la agrupación de almacenamiento ZFS resultante. Por ejemplo:

```
# zpool status
  pool: rpool
 state: ONLINE
 scrub: none requested
config:

        NAME          STATE     READ WRITE CKSUM
        rpool         ONLINE       0     0     0
          mirror-0    ONLINE       0     0     0
            c1t0d0s0  ONLINE       0     0     0
            c1t1d0s0  ONLINE       0     0     0

errors: No known data errors
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool              8.03G  58.9G    96K  /rpool
rpool/ROOT         4.47G  58.9G    21K  legacy
rpool/ROOT/zfsBE   4.47G  58.9G  4.47G  /
rpool/dump         1.50G  58.9G  1.50G  -
rpool/export         44K  58.9G    23K  /export
rpool/export/home    21K  58.9G    21K  /export/home
rpool/swap         2.06G  61.0G    16K  -
```

   La salida de `zfs list` de ejemplo identifica los componentes de la agrupación raíz, por ejemplo el directorio `rpool/ROOT`, al que de forma predeterminada no se puede acceder.

8. Si desea crear otro entorno de inicio ZFS en la misma agrupación de almacenamiento, puede utilizar el comando `lucreate`. En el ejemplo siguiente, se crea un nuevo entorno de inicio denominado `zfs2BE`. El entorno de inicio actual se denomina `zfsBE`, como se muestra en la salida `zfs list`. Sin embargo, el entorno de inicio actual no se confirma en la salida `lustatus` hasta que se crea el entorno de inicio nuevo.

```
# lustatus
ERROR: No boot environments are configured on this system
ERROR: cannot determine list of all boot environment names
```

EJEMPLO 5–1 Instalación inicial de un sistema de archivos raíz ZFS que se puede iniciar *(Continuación)*

Si se crea un entorno de inicio nuevo en la misma agrupación de inicio, se debe utilizar una sintaxis parecida a la siguiente:

```
# lucreate -n zfs2BE
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

La creación de un entorno de inicio ZFS en la misma agrupación utiliza las funciones de clonación e instantánea para crear instantáneamente el entorno de inicio. Si desea más información sobre el uso de Actualización automática de Oracle Solaris para una migración de raíz ZFS, consulte "Migración de un sistema de archivos raíz UFS a uno ZFS (Actualización automática de Oracle Solaris)" en la página 144.

9. A continuación, verifique los entornos de inicio nuevos. Por ejemplo:

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      yes    yes       no     -
zfs2BE                     yes      no     no        yes    -
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    8.03G  58.9G    97K  /rpool
rpool/ROOT               4.47G  58.9G    21K  legacy
rpool/ROOT/zfs2BE         116K  58.9G  4.47G  /
rpool/ROOT/zfsBE         4.47G  58.9G  4.47G  /
rpool/ROOT/zfsBE@zfs2BE  75.5K      -  4.47G  -
rpool/dump               1.50G  58.9G  1.50G  -
rpool/export               44K  58.9G    23K  /export
rpool/export/home          21K  58.9G    21K  /export/home
rpool/swap               2.06G  61.0G    16K  -
```

10. Para iniciar desde un entorno de inicio alternativo, use el comando `luactivate`. Después de activar el entorno de inicio en un sistema basado en SPARC, utilice el comando `boot -L` para identificar los entornos de inicio disponibles cuando el dispositivo de inicio contenga una agrupación de almacenamiento ZFS. En el caso de iniciar desde un sistema basado en x86, identifique el entorno de inicio se debe iniciar desde el menú GRUB.

EJEMPLO 5–1 Instalación inicial de un sistema de archivos raíz ZFS que se puede iniciar *(Continuación)*

Por ejemplo, en un sistema basado en SPARC, utilice el comando `boot -L` para obtener una lista con los entornos de inicio disponibles. Para iniciar desde el nuevo entorno de inicio, `zfs2BE`, seleccione la opción 2. A continuación, escriba el comando `boot -Z` que aparece.

```
ok boot -L
Executing last command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0  File and args: -L
1 zfsBE
2 zfs2BE
Select environment to boot: [ 1 - 2 ]: 2

To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfs2BE
ok boot -Z rpool/ROOT/zfs2BE
```

Si desea más información sobre cómo iniciar un sistema de archivos ZFS, consulte "Inicio desde un sistema de archivos raíz ZFS" en la página 170.

▼ Cómo crear una agrupación raíz reflejada (posterior a la instalación)

Si no creó una agrupación raíz ZFS reflejada durante la instalación, puede crear una fácilmente después de la instalación.

Si necesita información sobre la sustitución de un disco en una agrupación raíz, consulte "Cómo sustituir un disco en la agrupación raíz ZFS" en la página 178.

1 Muestre el estado actual de la agrupación raíz.

```
# zpool status rpool
  pool: rpool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        rpool       ONLINE       0     0     0
          c1t0d0s0  ONLINE       0     0     0

errors: No known data errors
```

2 Conecte un segundo disco para configurar una agrupación raíz reflejada.

```
# zpool attach rpool c1t0d0s0 c1t1d0s0
Please be sure to invoke installboot(1M) to make 'c1t1d0s0' bootable.
Make sure to wait until resilver is done before rebooting.
```

3 Vea el estado de la agrupación raíz para confirmar que se ha completado creación.

```
# zpool status rpool
  pool: rpool
 state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
 scrub: resilver in progress for 0h1m, 24.26% done, 0h3m to go
config:

        NAME          STATE     READ WRITE CKSUM
        rpool         ONLINE       0     0     0
          mirror-0    ONLINE       0     0     0
            c1t0d0s0  ONLINE       0     0     0
            c1t1d0s0  ONLINE       0     0     0  3.18G resilvered

errors: No known data errors
```

En la salida anterior, el proceso de creación no se ha completado. La creación se ha completado cuando se muestran mensajes parecidos al siguiente:

```
scrub: resilver completed after 0h10m with 0 errors on Thu Mar 11 11:27:22 2010
```

4 Aplique bloques de inicio al segundo disco cuando se haya completado la creación.

```
sparc# installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk /dev/rdsk/c1t1d0s0

x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1t1d0s0
```

5 Compruebe que puede iniciar desde el segundo disco.

6 Configure el sistema para que se inicie automáticamente desde el disco nuevo, mediante el comando `eeprom` o el comando `setenv` desde la PROM de inicio de SPARC. O bien, vuelva a configurar el BIOS del PC.

Instalación de un sistema de archivos raíz ZFS (instalación de archivo de almacenamiento flash de Oracle Solaris)

A partir de la versión Solaris 10 10/09, puede crear un archivo de almacenamiento flash en un sistema que esté ejecutando un sistema de archivos raíz UFS o un sistema de archivos raíz ZFS. Un archivo de almacenamiento flash de una agrupación ZFS contiene toda la jerarquía de la agrupación, excepto los volúmenes de intercambio y volcado, así como cualquier conjunto de datos excluido. Los volúmenes de intercambio y volcado se crean cuando se instala el archivo de almacenamiento flash. Puede utilizar el método de instalación del archivo de almacenamiento flash como sigue:

- Genere un archivo de almacenamiento flash que pueda utilizarse para instalar e iniciar un sistema con un sistema de archivos raíz ZFS.

- Realice una instalación JumpStart de un sistema mediante un archivo de almacenamiento flash ZFS. La creación de un archivo de almacenamiento flash ZFS clona toda una agrupación raíz, no entornos de inicio individuales. Los conjuntos de datos individuales en la agrupación se pueden excluir mediante la opción `D` del comando `flarcreate` y el comando `-flar`.

Revise las siguientes limitaciones antes de considerar la instalación de un sistema con un archivo de almacenamiento flash ZFS:

- Sólo se admite una instalación JumpStart de un archivo de almacenamiento flash ZFS. No se puede usar la opción de instalación interactiva de un archivo de almacenamiento flash para instalar un sistema con un sistema de archivos raíz ZFS. Tampoco puede utilizar un archivo de almacenamiento flash para instalar un entorno de inicio ZFS con Actualización automática de Oracle Solaris.
- Sólo puede instalar un archivo de almacenamiento flash en un sistema que tenga la misma arquitectura que el sistema en el que se creó el archivo de almacenamiento flash ZFS. Por ejemplo, un archivo de almacenamiento que se haya creado en un sistema `sun4u` no se puede instalar en un sistema `sun4v`.
- Sólo se admite una instalación inicial completa de un archivo de almacenamiento flash ZFS. No es posible instalar un archivo de almacenamiento flash diferencial de un sistema de archivos raíz ZFS ni un archivo de almacenamiento UFS/ZFS híbrido.
- Los archivos de almacenamiento flash UFS existentes sólo se pueden seguir utilizando para instalar un sistema de archivos raíz UFS. El archivo de almacenamiento flash ZFS sólo puede utilizarse para instalar un sistema de archivos raíz ZFS.
- Aunque toda la agrupación raíz, menos los conjuntos de datos explícitamente excluidos, esté archivada e instalada, sólo se puede utilizar el entorno de inicio ZFS que se inicie durante la creación del archivo de almacenamiento, después de que se instale el archivo de almacenamiento flash. Sin embargo, las agrupaciones que se han archivado con la opción `-R` del comando `flarcreate` o `flar` *rootdir* se pueden usar para archivar una agrupación raíz que no sea la que se ha iniciado actualmente.
- El nombre de la agrupación raíz ZFS que se crea con un archivo de almacenamiento flash debe coincidir con el nombre de la agrupación raíz principal. El nombre de la agrupación raíz que se usa para crear el archivo de almacenamiento flash es el nombre que se asigna a la nueva agrupación creada. El cambio de nombre de la agrupación no se admite.
- Las opciones de los comandos `flarcreate` y `flar` para incluir y excluir archivos individuales no se admiten en un archivo de almacenamiento flash ZFS. Sólo se pueden excluir conjuntos de datos completos desde un archivo de almacenamiento flash ZFS.
- El comando `flar info` no se admite para un archivo de almacenamiento flash ZFS. Por ejemplo:

```
# flar info -l zfs10u8flar
ERROR: archive content listing not supported for zfs archives.
```

Después de que se haya instalado al menos Solaris 10 10/09 en el sistema principal o se haya actualizado a dicha versión, puede crear un archivo de almacenamiento flash ZFS a fin de utilizarlo para instalar un sistema de destino. A continuación se expone el proceso básico:

- Instale o actualice al menos a la versión Solaris 10 10/09 en el sistema principal. Agregue las personalizaciones que desee.
- Cree el archivo de almacenamiento flash ZFS con el comando `flarcreate` en el sistema principal. Todos los conjuntos de datos de la agrupación raíz, excepto para los volúmenes de intercambio y volcado, se incluyen en el archivo de almacenamiento flash ZFS.
- Cree un perfil de JumpStart para que incluya la información del archivo de almacenamiento flash en el servidor de instalación.
- Instale el archivo de almacenamiento flash ZFS en el sistema de destino.

Las siguientes opciones de archivo de almacenamiento son compatibles para instalar una agrupación raíz ZFS con un archivo de almacenamiento flash:

- Utilice el comando `flarcreate` o `flar` para crear un archivo de almacenamiento flash desde la agrupación raíz ZFS especificada. Si no se especifica, se crea un archivo de almacenamiento flash de la agrupación raíz predeterminada.
- Utilice `flarcreate -D` *conjunto_datos* para excluir los conjuntos de datos especificados del archivo de almacenamiento flash. Esta opción se puede usar varias veces para excluir varios conjuntos de datos.

Después de instalar un archivo de almacenamiento flash ZFS, el sistema se configura como sigue:

- Toda la jerarquía del conjunto de datos que existía en el sistema en el que se creó el archivo de almacenamiento flash se vuelve a crear en el sistema de destino, menos los conjuntos de datos que se excluyeron específicamente en el momento de creación del archivo de almacenamiento. Los volúmenes de intercambio y volcado no se incluyen en el archivo de almacenamiento flash.
- La agrupación raíz tiene el mismo nombre que la agrupación que se usó para crear el archivo de almacenamiento.
- El entorno de inicio que estaba activo en el momento en el que se creó el archivo de almacenamiento flash es el entorno de inicio activo y predeterminado en los sistemas implementados.

EJEMPLO 5–2 Instalación de un sistema con un archivo de almacenamiento flash ZFS

Después de que se haya instalado al menos Solaris 10 10/09 en el sistema principal o se haya actualizado a dicha versión, cree un archivo de almacenamiento flash de la agrupación raíz ZFS. Por ejemplo:

```
# flarcreate -n zfsBE zfs10upflar
Full Flash
Checking integrity...
```

EJEMPLO 5–2 Instalación de un sistema con un archivo de almacenamiento flash ZFS *(Continuación)*

```
Integrity OK.
Running precreation scripts...
Precreation scripts done.
Determining the size of the archive...
The archive will be approximately 4.94GB.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.

Running pre-exit scripts...
Pre-exit scripts done.
```

En el sistema que se utilizará como servidor de instalación, cree un perfil JumpStart como lo haría para instalar cualquier sistema. Por ejemplo, el siguiente perfil se usa para instalar el archivo de almacenamiento `zfs10upflar`.

```
install_type flash_install
archive_location nfs system:/export/jump/zfs10upflar
partitioning explicit
pool rpool auto auto auto mirror c0t1d0s0 c0t0d0s0
```

Instalación de un sistema de archivos raíz ZFS (instalación JumpStart de Oracle Solaris)

Puede crear un perfil JumpStart para instalar un sistema de archivos raíz ZFS o un sistema de archivos raíz UFS.

Un perfil propio de ZFS debe contener la nueva palabra clave `pool`. La palabra clave `pool` instala una nueva agrupación raíz y, de forma predeterminada, se crea un nuevo entorno de inicio. Puede proporcionar el nombre del entorno de inicio y crear un conjunto de datos `/var` aparte con las palabras clave `bootenv installbe` y las opciones `bename` y `dataset`.

Para obtener información general sobre el uso de las funciones de JumpStart, consulte *Guía de instalación de Oracle Solaris 10 9/10: Instalaciones JumpStart personalizadas y avanzadas*.

Si va a configurar las zonas después de la instalación JumpStart de un sistema de archivos raíz ZFS y tiene previsto aplicar parches o actualizaciones al sistema, consulte "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)" en la página 151 o "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156.

Palabras clave de JumpStart para ZFS

Las siguientes palabras clave se permiten en un perfil propio de ZFS:

`auto` Especifica automáticamente el tamaño de los segmentos para la agrupación, el volumen de intercambio o el de volcado. Se comprueba el tamaño del disco para verificar que tenga cabida el tamaño mínimo. Si tiene cabida el tamaño mínimo, el tamaño máximo de agrupación se asigna según las limitaciones, por ejemplo el tamaño de los discos, los segmentos que se mantienen, etcétera.

Por ejemplo, si se especifica `c0t0d0s0`, se crea el segmento de agrupación raíz con el tamaño máximo al especificar las palabras clave `all` o `auto`. También puede especificarse un determinado tamaño para el segmento o el volumen de intercambio o volcado.

La palabra clave `auto` funciona de forma parecida a `all` si se utiliza con una agrupación raíz ZFS porque las agrupaciones carecen del concepto de espacio no utilizado.

`bootenv` Identifica las características del entorno de inicio.

Utilice la siguiente sintaxis de la palabra clave `bootenv` para crear un entorno raíz ZFS que se pueda iniciar:

`bootenv installbe bename` *nombre_entorno_inicio* [*conjunto_datos punto_montaje*]

`installbe`	Crea un entorno de inicio que se identifica mediante la opción `bename` y la entrada de *nombre_entorno_inicio*, y lo instala.
`bename` *nombre_entorno_inicio*	Identifica el *nombre_entorno_inicio* que se va a instalar. Si `bename` no se utiliza con la palabra clave `pool`, se crea un entorno de inicio predeterminado.
`dataset` *punto_montaje*	Utilice la palabra clave opcional `dataset` para identificar un conjunto de datos de `/var` independiente del conjunto de datos raíz. El valor de *punto_montaje* actualmente se limita a `/var`. Por ejemplo, una línea de sintaxis `bootenv` para un conjunto de datos de `/var` sería similar a lo siguiente:

```
bootenv installbe bename zfsroot dataset /var
```

`pool` Define la nueva agrupación raíz que se va a crear. Se debe proporcionar la siguiente sintaxis de palabra clave:

`pool` *poolname poolsize swapsize dumpsize vdevlist*

nombre_agrupación	Identifica el nombre de la agrupación que se va a crear. La agrupación se crea con el *tamaño* de agrupación indicado y con los dispositivos físicos especificados (*vdisp*). El valor `poolname` no debe identificar el nombre de una agrupación que exista o dicha agrupación se sobrescribirá.
tamaño_agrupación	Especifica el tamaño de la agrupación que se va a crear. El valor puede ser `auto` o `existing`. El valor `auto` asigna el máximo tamaño de agrupación posible según limitaciones como el tamaño de los discos, los segmentos que se mantienen, etcétera. El valor `existing` significa que los límites de los segmentos existentes según ese nombre se mantienen y no se sobrescriben. A menos que indique `g` (gigabytes), se da por sentado que el tamaño es en megabytes.
tamaño_intercambio	Especifica el tamaño del volumen de intercambio que se va a crear. El valor `auto` significa que se utiliza el tamaño de intercambio predeterminado. Puede especificar un tamaño con un valor *tamaño*. El tamaño es en MB, a menos que lo especifique por `g` (GB).
tamaño_volcado	Especifica el tamaño del volumen de volcado que se va a crear. El valor `auto` significa que se utiliza el valor tamaño de intercambio. Puede especificar un tamaño con un valor *tamaño*. A menos que indique `g` (gigabytes), se da por sentado que el tamaño es en megabytes.
lista_dispositivos_volumen	Especifica uno o más dispositivos que se utilizan para crear la agrupación. El formato de *lista_dispositivos_volumen* es el mismo que el del comando `zpool create`. Hasta el momento, las configuraciones reflejadas sólo son factibles si se especifican varios dispositivos. Los dispositivos de la *lista_dispositivos_volumen* deben ser segmentos de la agrupación raíz. El valor `any` significa que el software de instalación selecciona un dispositivo apropiado. Puede reflejar cuantos discos quiera. Ahora bien, el tamaño de la agrupación que se crea queda determinado por el disco más pequeño de todos los

discos que se especifiquen. Si desea más información sobre cómo crear agrupaciones de almacenamiento reflejadas, consulte "Configuración reflejada de agrupaciones de almacenamiento" en la página 69.

Ejemplos de perfil JumpStart ZFS

En esta sección se proporcionan ejemplos de perfiles JumpStart propios de ZFS.

El perfil siguiente efectúa una instalación inicial especificada con `install_type` *instalación_inicial* en una agrupación nueva, identificada con `pool` *agrupación_nueva*, cuyo tamaño se establece automáticamente mediante la palabra clave `auto` en el tamaño de los discos especificados. De manera automática, se asigna un tamaño al área de intercambio y el dispositivo de volcado mediante la palabra clave `auto` en una configuración reflejada de discos (con la palabra clave `mirror` y los discos especificados como *c0t0d0s0* y *c0t1d0s0*). Las características del entorno de inicio se establecen con la palabra clave `bootenv` para instalar un nuevo entorno de inicio con la palabra clave `installbe` y se crea un `bename` denominado *s10-xx*.

```
install_type initial_install
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename s10-xx
```

El perfil siguiente efectúa una instalación inicial con la palabra clave `install_type` *instalación_inicial* del metaclúster `SUNWCall` en una agrupación nueva denominada `newpool` que tiene un tamaño de 80 GB. Esta agrupación se crea con un volumen de intercambio de 2 GB y un volumen de volcado de 2 GB, en una configuración reflejada de dos dispositivos suficientemente grandes como para crear una agrupación de 80 GB. La instalación no puede realizarse correctamente si esos dos dispositivos no están disponibles. Las características del entorno de inicio se establecen con la palabra clave `bootenv` para instalar un nuevo entorno de inicio con la palabra clave `installbe` y se crea un `bename` denominado *s10–xx*.

```
install_type initial_install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename s10-xx
```

La sintaxis de instalación de JumpStart admite la capacidad de mantener o crear un sistema de archivos UFS en un disco que también incluya una agrupación raíz ZFS. Esta configuración no se recomienda en sistemas de producción; sin embargo, es apta para una transición o migración en un sistema pequeño, por ejemplo un portátil.

Problemas de JumpStart para ZFS

Antes de comenzar una instalación JumpStart en un sistema de archivos raíz ZFS que se puede iniciar, tenga en cuenta los problemas siguientes:

- Para crear un sistema de archivos raíz ZFS que se puede iniciar no se puede utilizar una agrupación de almacenamiento de una instalación JumpStart. Se debe crear una agrupación de almacenamiento ZFS con una sintaxis similar a la siguiente:

  ```
  pool rpool 20G 4G 4G c0t0d0s0
  ```

- Debe crear una agrupación con segmentos de disco en lugar de discos enteros, como se explica en "Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS" en la página 127. Por ejemplo, la sintaxis en negrita en el siguiente ejemplo no es aceptable:

  ```
  install_type initial_install
  cluster SUNWCall
  pool rpool all auto auto mirror c0t0d0 c0t1d0
  bootenv installbe bename newBE
  ```

  La sintaxis en negrita en el ejemplo siguiente es aceptable:

  ```
  install_type initial_install
  cluster SUNWCall
  pool rpool all auto auto mirror c0t0d0s0 c0t1d0s0
  bootenv installbe bename newBE
  ```

Migración de un sistema de archivos raíz UFS a uno ZFS (Actualización automática de Oracle Solaris)

Las funciones de Actualización automática de Oracle Solaris relacionadas con componentes UFS siguen disponibles, y funcionan igual que en las versiones anteriores de Solaris.

También están disponibles estas funciones:

- Al migrar un sistema de archivos raíz UFS a uno ZFS, se debe designar una agrupación de almacenamiento ZFS que ya exista con la opción `-p`.
- Si el sistema de archivos raíz UFS tiene componentes en distintos segmentos, se migran a la agrupación raíz ZFS.
- Puede migrar un sistema con zonas pero las configuraciones admitidas están limitadas en la versión Solaris 10 10/08. Se admiten más configuraciones de zona a partir de la versión Solaris 10 5/09. Para obtener más información, consulte las secciones siguientes:
  - "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)" en la página 151

- "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156

Si va a migrar un sistema sin zonas, consulte "Uso de Actualización automática de Oracle Solaris para migrar a un sistema de archivos raíz ZFS (sin zonas)" en la página 147.

- Actualización automática de Oracle Solaris puede utilizar la instantánea de ZFS y clonar funciones si se crea un entorno de inicio ZFS en la misma agrupación. Así, la creación de entornos de inicio es mucho más rápida que en versiones anteriores de Solaris.

Si desea más información sobre la instalación de Oracle Solaris y las funciones de Actualización automática de Oracle Solaris, consulte *Guía de instalación de Oracle Solaris 10 9/10: Actualización automática de Solaris y planificación de la actualización*.

A continuación se expone el procedimiento básico para migrar un sistema de archivos raíz UFS a uno ZFS:

- Instale la versión 10 10/08, Solaris 10 5/09, Solaris 10 10/09 o Oracle Solaris 10 9/10, o bien emplee un programa de actualización estándar para actualizar de una versión anterior de Solaris 10 en cualquier sistema admitido que se base en SPARC o x 86.
- Si se ejecuta al menos la versión Solaris 10 10/08, cree una agrupación de almacenamiento ZFS para el sistema de archivos raíz ZFS.
- Utilice Actualización automática de Oracle Solaris para migrar de un sistema de archivos raíz UFS a uno ZFS.
- Active el entorno de inicio ZFS con el comando `luactivate`.

Si necesita información sobre los requisitos de ZFS y Actualización automática de Oracle Solaris, consulte "Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS" en la página 127.

Problemas de migración ZFS con Actualización automática de Oracle Solaris

Antes de utilizar Actualización automática de Oracle Solaris para migrar el sistema de archivos UFS a un sistema de archivos raíz ZFS, examine la siguiente lista de problemas:

- La opción de actualización estándar de la GUI de Actualización automática de Oracle Solaris no está disponible para migrar de un sistema de archivos raíz UFS a uno ZFS. Para migrar de un sistema de archivos raíz UFS, debe utilizar Actualización automática de Oracle Solaris.
- Debe crear la agrupación de almacenamiento ZFS que se utilizará para el inicio antes de ejecutar Actualización automática de Oracle Solaris. Asimismo, debido a las actuales limitaciones de inicio, la agrupación raíz ZFS se debe crear con segmentos en lugar de discos enteros. Por ejemplo:

  ```
  # zpool create rpool mirror c1t0d0s0 c1t1d0s0
  ```

  Antes de crear la agrupación, compruebe que los discos que se usarán en ella tengan una etiqueta SMI (VTOC) en lugar de una etiqueta EFI. Si se vuelve a etiquetar el disco con una etiqueta SMI, compruebe que el proceso de etiquetado no haya modificado el esquema de partición. En la mayoría de los casos, toda la capacidad del disco debe estar en los segmentos que se destinan a la agrupación raíz.
- Actualización automática de Oracle Solaris no es apta para crear un entorno de inicio UFS a partir de un entorno de inicio ZFS. Si se migra el entorno de inicio UFS a uno ZFS y se mantiene el entorno de inicio UFS, se puede iniciar desde cualquiera de los dos entornos.
- No cambie el nombre de los entornos de inicio ZFS con el comando `zfs rename`, ya que Actualización automática de Oracle Solaris no detecta el cambio de nombre. Los comandos que se puedan usar posteriormente, por ejemplo `ludelete`, no funcionarán. De hecho, no cambie el nombre de agrupaciones ni de sistemas de archivos ZFS si tiene entornos de inicio que quiere seguir utilizando.
- Si se crea un entorno de inicio alternativo que sea un clon del entorno de inicio principal, no se pueden utilizar las opciones `-f`, `-x`, `-y`, `-Y` y `-z` para incluir o excluir archivos del entorno de inicio principal. Sin embargo, la opción de inclusión y exclusión se puede utilizar en los casos siguientes:

  ```
  UFS -> UFS
  UFS -> ZFS
  ZFS -> ZFS (different pool)
  ```
- Si bien Actualización automática de Oracle Solaris se puede usar para actualizar el sistema de archivos raíz UFS a uno ZFS, no es apta para la actualización de sistemas de archivos compartidos o que no sean raíz.
- El comando `lu` no es válido para crear o migrar un sistema de archivos raíz ZFS.

Uso de Actualización automática de Oracle Solaris para migrar a un sistema de archivos raíz ZFS (sin zonas)

Los ejemplos siguientes ilustran la manera de migrar un sistema de archivos raíz UFS a uno ZFS.

Si desea migrar o actualizar un sistema con zonas, consulte las siguientes secciones:

- "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)" en la página 151
- "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156

EJEMPLO 5–3 Uso de Actualización automática de Oracle Solaris para migrar un sistema de archivos raíz UFS a uno ZFS

El ejemplo siguiente ilustra la forma de crear un entorno de inicio de un sistema de archivos raíz ZFS desde un sistema de archivos raíz UFS. El entorno de inicio actual, `ufsBE`, que contiene un sistema de archivos raíz UFS, se identifica mediante la opción `-c`. Si no incluye la opción `-c` opcional, el nombre del entorno de inicio actual se convierte de forma predeterminada en el nombre del dispositivo. El entorno de inicio nuevo, `zfsBE`, se identifica mediante la opción `-n`. Antes de la operación `lucreate` debe haber una agrupación de almacenamiento ZFS.

Para que se pueda iniciar y actualizar, la agrupación de almacenamiento ZFS se debe crear con segmentos en lugar de discos enteros. Antes de crear la agrupación, compruebe que los discos que se usarán en ella tengan una etiqueta SMI (VTOC) en lugar de una etiqueta EFI. Si se vuelve a etiquetar el disco con una etiqueta SMI, compruebe que el proceso de etiquetado no haya modificado el esquema de partición. En la mayoría de los casos, toda la capacidad del disco debe estar en los segmentos que se destinan a la agrupación raíz.

```
# zpool create rpool mirror c1t2d0s0 c2t1d0s0
# lucreate -c ufsBE -n zfsBE -p rpool
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <ufsBE>.
Creating initial configuration for primary boot environment <ufsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c1t2d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
```

EJEMPLO 5–3 Uso de Actualización automática de Oracle Solaris para migrar un sistema de archivos raíz UFS a uno ZFS *(Continuación)*

```
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Populating file systems on boot environment <zfsBE>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Creating compare databases for boot environment <zfsBE>.
Creating compare database for file system </rpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfsBE>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-qD.mnt
updating /.alt.tmp.b-qD.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
```

Tras finalizar la operación de `lucreate`, utilice el comando `lustatus` para ver el estado del entorno de inicio. Por ejemplo:

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
ufsBE                      yes      yes    yes       no     -
zfsBE                      yes      no     no        yes    -
```

A continuación, examine la lista de componentes de ZFS. Por ejemplo:

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
rpool                 7.17G  59.8G  95.5K  /rpool
rpool/ROOT            4.66G  59.8G    21K  /rpool/ROOT
rpool/ROOT/zfsBE      4.66G  59.8G  4.66G  /
rpool/dump               2G  61.8G    16K  -
rpool/swap             517M  60.3G    16K  -
```

Después, utilice el comando `luactivate` para activar el nuevo entorno de inicio ZFS. Por ejemplo:

```
# luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.

**********************************************************************

The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.
```

EJEMPLO 5–3 Uso de Actualización automática de Oracle Solaris para migrar un sistema de archivos raíz UFS a uno ZFS *(Continuación)*

```
**********************************************************************
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
```

A continuación, reinicie el sistema en el entorno de inicio ZFS.

```
# init 6
```

Confirme que el entorno de inicio ZFS esté activo.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
ufsBE                      yes      no     no        yes    -
zfsBE                      yes      yes    yes       no     -
```

Si vuelve al entorno de inicio UFS, tendrá que volver a importar todas las agrupaciones de almacenamiento ZFS creadas en el entorno de inicio ZFS porque no están disponibles automáticamente en el entorno de inicio UFS.

Si ya no se necesita el entorno de inicio UFS, se puede eliminar con el comando `ludelete`.

EJEMPLO 5–4 Uso de Actualización automática de Oracle Solaris para crear un entorno de inicio ZFS a partir de un entorno de inicio ZFS

El proceso de creación de un entorno de inicio ZFS desde un entorno de inicio ZFS es muy rápido porque esta operación utiliza las funciones de clonación e instantánea de ZFS. Si el entorno de inicio actual reside en la misma agrupación ZFS, se omite la opción `-p`.

Si tiene varios entornos de inicio ZFS, lleve a cabo el siguiente procedimiento para seleccionar el entorno de inicio desde el que desea iniciar:

- SPARC: puede utilizar el comando `boot -L` para identificar los entornos de inicio disponibles y seleccionar un entorno de inicio desde el que se efectúe el inicio mediante el comando `boot -Z`.
- x86: puede seleccionar un entorno de inicio desde el menú GRUB.

Para obtener más información, consulte el Ejemplo 5–9.

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
```

EJEMPLO 5–4 Uso de Actualización automática de Oracle Solaris para crear un entorno de inicio ZFS a partir de un entorno de inicio ZFS *(Continuación)*

```
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

EJEMPLO 5–5 Actualización del entorno de inicio ZFS (`luupgrade`)

El entorno de inicio ZFS se puede actualizar con paquetes o parches adicionales.

A continuación se expone el proceso básico:

- Crear un entorno de inicio alternativo con el comando `lucreate`.
- Activar e iniciar desde el entorno de inicio alternativo.
- Actualizar el entorno de inicio ZFS principal con el comando `luupgrade` para agregar paquetes o parches.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      no     no        yes    -
zfs2BE                     yes      yes    yes       no     -
# luupgrade -p -n zfsBE -s /net/system/export/s10up/Solaris_10/Product SUNWchxge
Validating the contents of the media </net/install/export/s10up/Solaris_10/Product>.
Mounting the BE <zfsBE>.
Adding packages to the BE <zfsBE>.

Processing package instance <SUNWchxge> from </net/install/export/s10up/Solaris_10/Product>

Chelsio N110 10GE NIC Driver(sparc) 11.10.0,REV=2006.02.15.20.41
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.

This appears to be an attempt to install the same architecture and
version of a package which is already installed.  This installation
will attempt to overwrite this package.

Using </a> as the package base directory.
## Processing package information.
```

EJEMPLO 5–5 Actualización del entorno de inicio ZFS (`luupgrade`) *(Continuación)*

```
## Processing system information.
   4 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SUNWchxge> [y,n,?] y
Installing Chelsio N110 10GE NIC Driver as <SUNWchxge>

## Installing part 1 of 1.
## Executing postinstall script.

Installation of <SUNWchxge> was successful.
Unmounting the BE <zfsBE>.
The package add to the BE <zfsBE> completed.
```

Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)

Actualización automática de Oracle Solaris se puede utilizar para migrar un sistema con zonas, pero las configuraciones admitidas son limitadas en la versión Solaris 10 10/08. Si está instalando o actualizando al menos a la versión Solaris 10 5/09, se admiten más configuraciones de zona. Para obtener más información, consulte "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156.

En esta sección se explica el procedimiento para configurar e instalar un sistema con zonas para poder actualizarlo y aplicarle parches mediante Actualización automática de Oracle Solaris. Si va a migrar a un sistema de archivos raíz ZFS sin zonas, consulte "Uso de Actualización automática de Oracle Solaris para migrar a un sistema de archivos raíz ZFS (sin zonas)" en la página 147.

Si va a migrar un sistema con zonas, o bien si tiene previsto configurar un sistema con zonas en la versión Solaris 10 10/08, consulte los procedimientos siguientes:

- "Cómo migrar un sistema de archivos raíz UFS con raíces de zona en UFS a un sistema de archivos raíz ZFS (Solaris 10 10/08)" en la página 152
- "Cómo configurar un sistema de archivos raíz ZFS con raíces de zona en ZFS (Solaris 10 10/08)" en la página 154

- "Cómo actualizar o aplicar parches a un sistema de archivos raíz ZFS con raíces de zona en ZFS (Solaris 10 10/08)" en la página 155
- "Resolución de problemas de punto de montaje ZFS que impiden un inicio correcto (Solaris 10 10/08)" en la página 175

Siga los procedimientos recomendados para configurar zonas de un sistema con un sistema de archivos raíz ZFS para asegurarse de poder utilizar en él Actualización automática de Oracle Solaris.

▼ Cómo migrar un sistema de archivos raíz UFS con raíces de zona en UFS a un sistema de archivos raíz ZFS (Solaris 10 10/08)

Este procedimiento explica cómo migrar un sistema de archivos raíz UFS con zonas instaladas a un sistema de archivos raíz ZFS y una configuración raíz de zona ZFS que se pueda actualizar o a la que se puedan aplicar parches.

En los pasos siguientes, el nombre de la agrupación de ejemplo es `rpool` y el del entorno de inicio actualmente activo es `s10BE*`.

1 Actualice el sistema a la versión Solaris 10 10/08 si se ejecuta una versión de Solaris 10 anterior.

Para obtener más información sobre cómo actualizar un sistema que ejecuta la versión Solaris 10, consulte *Guía de instalación de Oracle Solaris 10 9/10: Actualización automática de Solaris y planificación de la actualización*.

2 Cree la agrupación raíz.

```
# zpool create rpool mirror c0t1d0 c1t1d0
```

Si necesita información sobre los requisitos de agrupaciones raíz, consulte "Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS" en la página 127.

3 Confirme que se hayan iniciado las zonas desde el entorno de inicio UFS.

4 Cree el nuevo entorno de inicio ZFS.

```
# lucreate -n s10BE2 -p rpool
```

Este comando establece conjuntos de datos en la agrupación raíz del nuevo entorno de inicio y copia el entorno de inicio actual (zonas incluidas) en esos conjuntos de datos.

5 Active el nuevo entorno de inicio ZFS.

```
# luactivate s10BE2
```

El sistema ya ejecuta un sistema de archivos raíz ZFS; sin embargo, las raíces de zona de UFS siguen estando en el sistema de archivos raíz UFS. Los pasos siguientes son necesarios para migrar por completo las zonas UFS a una configuración ZFS compatible.

6 Reinicie el sistema.

```
# init 6
```

7 Migre las zonas a un entorno de inicio ZFS.

a. Inicie las zonas.

b. Cree otro entorno de inicio en la agrupación.

```
# lucreate s10BE3
```

c. Active el nuevo entorno de inicio.

```
# luactivate s10BE3
```

d. Reinicie el sistema.

```
# init 6
```

En este paso se verifica que se hayan iniciado el entorno de inicio ZFS y las zonas.

8 Solucione los posibles problemas de punto de montaje.

Debido a un error en Actualización automática de Oracle Solaris, el inicio del entorno de inicio no activo podría fallar porque un conjunto de datos ZFS o el conjunto de datos ZFS de una zona del entorno de inicio tiene un punto de montaje no válido.

a. Examine la salida de `zfs list`.

Busque puntos de montaje temporales incorrectos. Por ejemplo:

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10u6

NAME                                MOUNTPOINT
rpool/ROOT/s10u6                    /.alt.tmp.b-VP.mnt/
rpool/ROOT/s10u6/zones              /.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10u6/zones/zonerootA    /.alt.tmp.b-VP.mnt/zones/zonerootA
```

El punto de montaje del entorno de inicio ZFS (`rpool/ROOT/s10u6`) debe ser `/`.

b. Restablezca los puntos de montaje del entorno de inicio ZFS y sus conjuntos de datos.

Por ejemplo:

```
# zfs inherit -r mountpoint rpool/ROOT/s10u6
# zfs set mountpoint=/ rpool/ROOT/s10u6
```

c. Reinicie el sistema.

Si se presenta la opción para iniciar un determinado entorno de inicio, en el menú GRUB o el indicador de OpenBoot PROM, seleccione el entorno de inicio cuyos puntos de montaje se han acabado de corregir.

▼ Cómo configurar un sistema de archivos raíz ZFS con raíces de zona en ZFS (Solaris 10 10/08)

Este procedimiento explica cómo configurar un sistema de archivos raíz ZFS y una configuración raíz de zona ZFS que se pueda actualizar o a la que se pueda aplicar parches. En esta configuración, las raíces de zona ZFS se crean como conjuntos de datos ZFS.

En los pasos siguientes, el nombre de la agrupación de ejemplo es `rpool` y el del entorno de inicio actualmente activo es `s10BE`. Cualquier nombre de conjunto de datos legal es valido como nombre para el conjunto de datos de zonas. En el ejemplo siguiente, el nombre del conjunto de datos de las zonas es `zones`.

1 Instale el sistema con una raíz ZFS, ya sea con el método del instalador de texto interactivo de Solaris o con el de la instalación Solaris JumpStart.

Si desea información sobre cómo instalar un sistema de archivos raíz ZFS con el método de instalación inicial o con Solaris JumpStart, consulte "Instalación de un sistema de archivos raíz ZFS (instalación inicial)" en la página 130 o "Instalación de un sistema de archivos raíz ZFS (instalación JumpStart de Oracle Solaris)" en la página 140.

2 Inicie el sistema desde la agrupación raíz recién creada.

3 Cree un conjunto de datos para agrupar las raíces de zona.

Por ejemplo:

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones
```

El establecimiento del valor `noauto` para la propiedad `canmount` impide que otra acción distinta de la explícita de Actualización automática de Oracle Solaris y el código de inicio del sistema monte el conjunto de datos.

4 Monte el conjunto de datos de zonas recién creado.

```
# zfs mount rpool/ROOT/s10BE/zones
```

El conjunto de datos se monta en `/zones`.

5 Cree y monte un conjunto de datos para cada raíz de zona.

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones/zonerootA
# zfs mount rpool/ROOT/s10BE/zones/zonerootA
```

6 Establezca los permisos pertinentes en el directorio raíz de zona.

```
# chmod 700 /zones/zonerootA
```

7 Configure la zona estableciendo la ruta de zona como se indica a continuación:

```
# zonecfg -z zoneA
    zoneA: No such zone configured
    Use 'create' to begin configuring a new zone.
```

```
zonecfg:zoneA> create
zonecfg:zoneA> set zonepath=/zones/zonerootA
```

Puede habilitar las zonas para que se inicien automáticamente cuando se inicie el sistema mediante la sintaxis siguiente:

```
zonecfg:zoneA> set autoboot=true
```

8 Instale la zona.

```
# zoneadm -z zoneA install
```

9 Inicie la zona.

```
# zoneadm -z zoneA boot
```

▼ Cómo actualizar o aplicar parches a un sistema de archivos raíz ZFS con raíces de zona en ZFS (Solaris 10 10/08)

Utilice este procedimiento cuando deba actualizar o aplicar parches a un sistema de archivos raíz ZFS con raíces de zona en ZFS. Las actualizaciones pueden consistir en una actualización del sistema o la aplicación de parches.

En los pasos siguientes, newBE es el nombre de ejemplo del entorno de inicio que se actualiza o al que se aplican parches.

1 Cree el entorno de inicio para actualizar o al que aplicar parches.

```
# lucreate -n newBE
```

Se clona el entorno de inicio que ya existe, incluidas todas las zonas. Se crea un conjunto de datos para cada conjunto de datos del entorno de inicio original. Los nuevos conjuntos de datos se crean en la misma agrupación que la agrupación raíz actual.

2 Seleccione una de las opciones siguientes para actualizar el sistema o aplicar parches al nuevo entorno de inicio:

- Actualice el sistema.

  ```
  # luupgrade -u -n newBE -s /net/install/export/s10u7/latest
  ```

  La opción -s especifica la ubicación de un medio de instalación de Solaris.

- Aplique parches al nuevo entorno de inicio.

  ```
  # luupgrade -t -n newBE -t -s /patchdir 139147-02 157347-14
  ```

3 Active el nuevo entorno de inicio.

```
# luactivate newBE
```

4 **Inicie desde el entorno de inicio recién activado.**

```
# init 6
```

5 **Solucione los posibles problemas de punto de montaje.**

Debido a un error en la función de Actualización automática de Oracle Solaris, el inicio del entorno de inicio no activo podría fallar porque un conjunto de datos ZFS o el conjunto de datos ZFS de una zona del entorno de inicio tiene un punto de montaje no válido.

a. **Examine la salida de `zfs list`.**

Busque puntos de montaje temporales incorrectos. Por ejemplo:

```
# zfs list -r -o name,mountpoint rpool/ROOT/newBE

NAME                                MOUNTPOINT
rpool/ROOT/newBE                    /.alt.tmp.b-VP.mnt/
rpool/ROOT/newBE/zones              /.alt.tmp.b-VP.mnt/zones
rpool/ROOT/newBE/zones/zonerootA    /.alt.tmp.b-VP.mnt/zones/zonerootA
```

El punto de montaje del entorno de inicio ZFS raíz (`rpool/ROOT/newBE`) debe ser `/`.

b. **Restablezca los puntos de montaje del entorno de inicio ZFS y sus conjuntos de datos.**

Por ejemplo:

```
# zfs inherit -r mountpoint rpool/ROOT/newBE
# zfs set mountpoint=/ rpool/ROOT/newBE
```

c. **Reinicie el sistema.**

Si se presenta la opción para iniciar un determinado entorno de inicio, en el menú GRUB o el indicador de OpenBoot PROM, seleccione el entorno de inicio cuyos puntos de montaje se han acabado de corregir.

Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)

Puede usar la función Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas a partir de la versión Solaris 10 10/08. Actualización automática admite las configuraciones de zona completas y dispersas adicionales a partir de la versión Solaris 10 5/09.

En esta sección se describe cómo configurar un sistema con zonas para que se pueda aplicar un parche o una actualización con Actualización automática de Oracle Solaris a partir de la versión Solaris 10 5/09. Si va a migrar a un sistema de archivos raíz ZFS sin zonas, consulte "Uso de Actualización automática de Oracle Solaris para migrar a un sistema de archivos raíz ZFS (sin zonas)" en la página 147.

Tenga en cuenta los puntos siguientes cuando se utilice Actualización automática de Oracle Solaris con ZFS y zonas a partir de la versión Solaris 10 5/09:

- Para utilizar Actualización automática de Oracle Solaris con configuraciones de zona que se admiten a partir al menos de la versión Solaris 10 5/09, en primer lugar debe actualizar el sistema, al menos, a la versión Solaris 10 5/09, mediante el programa estándar de actualización.
- A continuación, mediante Actualización automática de Oracle Solaris, puede migrar el sistema de archivos raíz UFS con raíces de zona a un sistema de archivos raíz ZFS, o bien puede aplicar un parche o una actualización al sistema de archivos raíz ZFS y las raíces de zona.
- No se pueden migrar configuraciones de zona no admitidas de una versión anterior de Solaris 10 directamente a la versión Solaris 10 5/09.

Si está migrando o configurando un sistema con zonas a partir de la versión Solaris 10 5/09, revise la siguiente información:

- "ZFS admitido con información de configuración de raíces de zona (al menos Solaris 10 5/09) " en la página 157
- "Cómo crear un entorno de inicio ZFS con un sistema de archivos raíz ZFS y una raíz de zona (al menos Solaris 10 5/09) " en la página 159
- "Cómo aplicar un parche o una actualización a un sistema de archivos raíz ZFS con raíces de zona (al menos Solaris 10 5/09)" en la página 160
- "Cómo migrar un sistema de archivos raíz UFS con una raíz de zona a un sistema de archivos raíz ZFS (al menos Solaris 10 5/09)" en la página 164

ZFS admitido con información de configuración de raíces de zona (al menos Solaris 10 5/09)

Revise las configuraciones de zona admitidas antes de usar la función Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas.

- **Migración de un sistema de archivos raíz UFS a un sistema de archivos raíz ZFS** – Se admiten las siguientes configuraciones de raíces de zona:
  - En un directorio del sistema de archivos raíz UFS
  - En un subdirectorio de un punto de montaje en el sistema de archivos raíz UFS
  - Sistema de archivos raíz UFS con una raíz de zona en un directorio de sistema de archivos raíz UFS o en un subdirectorio de un punto de montaje de un sistema de archivos raíz UFS y una agrupación no raíz ZFS con una zona raíz

  La siguiente configuración UFS/zona no se admite: sistema de archivos raíz UFS que tiene una raíz de zona como punto de montaje.

- **Migración o actualización de un sistema de archivos raíz ZFS** – Se admiten las siguientes configuraciones de raíces de zona:
  - En un conjunto de datos de la agrupación raíz ZFS. En algunos casos, si no se proporciona un conjunto de datos de la raíz de zona antes de la utilización de Actualización automática de Oracle Solaris, ésta creará un conjunto de datos para la raíz de zona (`zoneds`).
  - En un subdirectorio del sistema de archivos raíz ZFS
  - En un conjunto de datos fuera del sistema de archivos raíz ZFS
  - En el subdirectorio de un conjunto de datos fuera del sistema de archivos raíz ZFS
  - En un conjunto de datos de una agrupación no raíz. En el ejemplo siguiente, `zonepool/zones` es un conjunto de datos que contiene las raíces de zona y `rpool` contiene el entorno de inicio ZFS:

    ```
    zonepool
    zonepool/zones
    zonepool/zones/myzone
    rpool
    rpool/ROOT
    rpool/ROOT/myBE
    ```

    Actualización automática de Oracle Solaris toma una instantánea y clona las zonas de `zonepool` y el entorno de inicio `rpool` si utiliza esta sintaxis:

    ```
    # lucreate -n newBE
    ```

    Se crea el entorno de inicio `newBE` en `rpool/ROOT/newBE`. Si está activado, `newBE` proporciona acceso a los componentes de `zonepool`.

    En el ejemplo anterior, si `/zonepool/zones` fuera un subdirectorio, y no un conjunto de datos independiente, Actualización automática los migraría como componentes de la agrupación raíz, `rpool`.
- **Información de actualización o migración de zonas con zonas para UFS y ZFS** – Revise las siguientes consideraciones que pueden afectar a una migración o una actualización de un entorno ZFS y UFS:
  - Si ha configurado las zonas, tal y como se describe en "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)" en la página 151 en la versión Solaris 10 10/08 y ha actualizado al menos a Solaris 10 5/09, debería poder migrar a un sistema de archivos raíz ZFS o utilizar Actualización automática de Oracle Solaris para actualizar al menos a la versión Solaris 10 5/09.
  - No cree raíces de zona en directorios anidados, por ejemplo, `zones/zone1` y `zones/zone1/zone2`. De lo contrario, el montaje puede fallar en el momento del inicio.

▼ Cómo crear un entorno de inicio ZFS con un sistema de archivos raíz ZFS y una raíz de zona (al menos Solaris 10 5/09)

Utilice este procedimiento después de haber realizado una instalación inicial de, al menos, la versión Solaris 10 5/09 para crear un sistema de archivos raíz ZFS. Utilice este procedimiento después de utilizar la función `luupgrade` para la actualización de un sistema de archivos raíz ZFS al menos a la versión Solaris 10 5/09. Se puede aplicar una actualización o un parche a un entorno de inicio ZFS que se cree mediante este procedimiento.

En los pasos que aparecen a continuación, el sistema Oracle Solaris 10 9/10 de ejemplo tiene un sistema de archivos raíz ZFS y un conjunto de datos raíz de zona en `/rpool/zones`. Se crea un entorno de inicio ZFS denominado `zfs2BE` al que se puede aplicar una actualización o un parche.

1 Revise los sistemas de archivos ZFS existentes.

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
rpool                 7.26G  59.7G    98K  /rpool
rpool/ROOT            4.64G  59.7G    21K  legacy
rpool/ROOT/zfsBE      4.64G  59.7G  4.64G  /
rpool/dump            1.00G  59.7G  1.00G  -
rpool/export            44K  59.7G    23K  /export
rpool/export/home       21K  59.7G    21K  /export/home
rpool/swap               1G  60.7G    16K  -
rpool/zones            633M  59.7G   633M  /rpool/zones
```

2 Asegúrese de que las zonas se hayan instalado e iniciado.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /rpool/zones                   native   shared
```

3 Cree el entorno de inicio ZFS.

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
```

```
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

4 Active el entorno de inicio ZFS.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      yes    yes       no     -
zfs2BE                     yes      no     no        yes    -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
# init 6
```

5 Confirme que las zonas y los sistemas de archivos ZFS se creen en el nuevo entorno de inicio.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                              7.38G  59.6G    98K  /rpool
rpool/ROOT                         4.72G  59.6G    21K  legacy
rpool/ROOT/zfs2BE                  4.72G  59.6G  4.64G  /
rpool/ROOT/zfs2BE@zfs2BE           74.0M      -  4.64G  -
rpool/ROOT/zfsBE                   5.45M  59.6G  4.64G  /.alt.zfsBE
rpool/dump                         1.00G  59.6G  1.00G  -
rpool/export                         44K  59.6G    23K  /export
rpool/export/home                    21K  59.6G    21K  /export/home
rpool/swap                            1G  60.6G    16K  -
rpool/zones                        17.2M  59.6G   633M  /rpool/zones
rpool/zones-zfsBE                   653M  59.6G   633M  /rpool/zones-zfsBE
rpool/zones-zfsBE@zfs2BE           19.9M      -   633M  -
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   - zfszone          installed  /rpool/zones                   native   shared
```

▼ Cómo aplicar un parche o una actualización a un sistema de archivos raíz ZFS con raíces de zona (al menos Solaris 10 5/09)

Utilice este procedimiento cuando deba aplicar parches o actualizaciones a un sistema de archivos raíz ZFS con raíces de zona en la versión Solaris 10 5/09. Las actualizaciones pueden consistir en una actualización del sistema o la aplicación de parches.

En los pasos siguientes, `zfs2BE` es el nombre de ejemplo del entorno de inicio al que se ha aplicado una actualización o parche.

1 Revise los sistemas de archivos ZFS existentes.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
```

```
rpool                          7.38G  59.6G   100K  /rpool
rpool/ROOT                     4.72G  59.6G    21K  legacy
rpool/ROOT/zfs2BE              4.72G  59.6G  4.64G  /
rpool/ROOT/zfs2BE@zfs2BE       75.0M      -  4.64G  -
rpool/ROOT/zfsBE               5.46M  59.6G  4.64G  /
rpool/dump                     1.00G  59.6G  1.00G  -
rpool/export                     44K  59.6G    23K  /export
rpool/export/home                21K  59.6G    21K  /export/home
rpool/swap                        1G  60.6G    16K  -
rpool/zones                    22.9M  59.6G   637M  /rpool/zones
rpool/zones-zfsBE               653M  59.6G   633M  /rpool/zones-zfsBE
rpool/zones-zfsBE@zfs2BE       20.0M      -   633M  -
```

2 Asegúrese de que las zonas se hayan instalado e iniciado.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   5 zfszone          running    /rpool/zones                   native   shared
```

3 Cree el entorno de inicio ZFS al que aplicar actualizaciones o parches.

```
# lucreate -n zfs2BE
Analyzing system configuration.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Creating snapshot for <rpool/zones> on <rpool/zones@zfs10092BE>.
Creating clone for <rpool/zones@zfs2BE> on <rpool/zones-zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

4 Seleccione una de las opciones siguientes para actualizar el sistema o aplicar parches al nuevo entorno de inicio:

- Actualice el sistema.

  ```
  # luupgrade -u -n zfs2BE -s /net/install/export/s10up/latest
  ```

  La opción `-s` especifica la ubicación de un medio de instalación de Solaris.

  Este proceso puede durar mucho tiempo.

  Para obtener un ejemplo completo del proceso `luupgrade`, consulte el Ejemplo 5–6.

- Aplique parches al nuevo entorno de inicio.

  ```
  # luupgrade -t -n zfs2BE -t -s /patchdir patch-id-02 patch-id-04
  ```

5 Active el nuevo entorno de inicio.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      yes    yes       no     -
zfs2BE                     yes      no     no        yes    -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
```

6 Inicie desde el entorno de inicio recién activado.

```
# init 6
```

Ejemplo 5–6 Actualización de un sistema de archivos raíz ZFS con una raíz de zona a un sistema de archivos raíz ZFS de Oracle Solaris 9 10/10

En este ejemplo, un entorno de inicio ZFS (`zfsBE`), creado en un sistema Solaris 10 10/09 con un sistema de archivos raíz ZFS y raíz de zona en una agrupación no raíz, se actualiza a la versión Oracle Solaris 10 9/10. Este proceso puede durar mucho tiempo. A continuación, el entorno de inicio actualizado (`zfs2BE`) se activa. Asegúrese de que las zonas se hayan instalado e iniciado antes de intentar la actualización.

En este ejemplo, la agrupación `zonepool`, el conjunto de datos `/zonepool/zones`, así como la zona `zfszone` se crean de este modo:

```
# zpool create zonepool mirror c2t1d0 c2t5d0
# zfs create zonepool/zones
# chmod 700 zonepool/zones
# zonecfg -z zfszone
zfszone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zfszone> create
zonecfg:zfszone> set zonepath=/zonepool/zones
zonecfg:zfszone> verify
zonecfg:zfszone> exit
# zoneadm -z zfszone install
cannot create ZFS dataset zonepool/zones: dataset already exists
Preparing to install zone <zfszone>.
Creating list of files to copy from the global zone.
Copying <8960> files to the zone.
.
.
.

# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /zonepool/zones                native   shared
```

```
# lucreate -n zfsBE
.
.
.
# luupgrade -u -n zfsBE -s /net/install/export/s10up/latest
40410 blocks
miniroot filesystem is <lofs>
Mounting miniroot at </net/system/export/s10up/latest/Solaris_10/Tools/Boot>
Validating the contents of the media </net/system/export/s10up/latest>.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains <Solaris> version <10>.
Constructing upgrade profile to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE <zfsBE>.
Determining packages to install or upgrade for BE <zfsBE>.
Performing the operating system upgrade of the BE <zfsBE>.
CAUTION: Interrupting this process may leave the boot environment unstable
or unbootable.
Upgrading Solaris: 100% completed
Installation of the packages from this media is complete.
Updating package information on boot environment <zfsBE>.
Package information successfully updated on boot environment <zfsBE>.
Adding operating system patches to the BE <zfsBE>.
The operating system patch installation is complete.
INFORMATION: The file </var/sadm/system/logs/upgrade_log> on boot
environment <zfsBE> contains a log of the upgrade operation.
INFORMATION: The file </var/sadm/system/data/upgrade_cleanup> on boot
environment <zfsBE> contains a log of cleanup operations required.
INFORMATION: Review the files listed above. Remember that all of the files
are located on boot environment <zfsBE>. Before you activate boot
environment <zfsBE>, determine if any additional system maintenance is
required or if additional media of the software distribution must be
installed.
The Solaris upgrade of the boot environment <zfsBE> is complete.
Installing failsafe
Failsafe install is complete.
# luactivate zfsBE
# init 6
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      no     no        yes    -
zfs2BE                     yes      yes    yes       no     -
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   - zfszone          installed  /zonepool/zones                native   shared
```

▼ Cómo migrar un sistema de archivos raíz UFS con una raíz de zona a un sistema de archivos raíz ZFS (al menos Solaris 10 5/09)

Utilice este procedimiento para migrar un sistema con un sistema de archivos raíz UFS y una raíz de zona al menos a la versión Solaris 10 5/09. A continuación, utilice Actualización automática de Oracle Solaris para crear un entorno de inicio ZFS.

En los pasos que aparecen a continuación, el nombre del entorno de inicio UFS de ejemplo es `c0t1d0s0`, la raíz de zona UFS es `zonepool/zfszone` y el entorno de inicio raíz es `zfsBE`.

1 Actualice el sistema a la versión Solaris 10 5/09 si se ejecuta una versión de Solaris 10 anterior.

Para obtener información sobre cómo actualizar un sistema que ejecuta la versión Solaris 10, consulte *Guía de instalación de Oracle Solaris 10 9/10: Actualización automática de Solaris y planificación de la actualización*.

2 Cree la agrupación raíz.

Si necesita información sobre los requisitos de agrupaciones raíz, consulte "Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS" en la página 127.

3 Confirme que se hayan iniciado las zonas desde el entorno de inicio UFS.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /zonepool/zones                native   shared
```

4 Cree el nuevo entorno de inicio ZFS.

```
# lucreate -c c1t1d0s0 -n zfsBE -p rpool
```

Este comando establece conjuntos de datos en la agrupación raíz del nuevo entorno de inicio y copia el entorno de inicio actual (zonas incluidas) en esos conjuntos de datos.

5 Active el nuevo entorno de inicio ZFS.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
c1t1d0s0                   yes      no     no        yes    -
zfsBE                      yes      yes    yes       no     -         #
luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.
.
.
.
```

6 Reinicie el sistema.

```
# init 6
```

7 Confirme que las zonas y los sistemas de archivos ZFS se creen en el nuevo entorno de inicio.

```
# zfs list
NAME                              USED  AVAIL  REFER  MOUNTPOINT
rpool                            6.17G  60.8G    98K  /rpool
rpool/ROOT                       4.67G  60.8G    21K  /rpool/ROOT
rpool/ROOT/zfsBE                 4.67G  60.8G  4.67G  /
rpool/dump                       1.00G  60.8G  1.00G  -
rpool/swap                        517M  61.3G    16K  -
zonepool                          634M  7.62G    24K  /zonepool
zonepool/zones                    270K  7.62G   633M  /zonepool/zones
zonepool/zones-c1t1d0s0           634M  7.62G   633M  /zonepool/zones-c1t1d0s0
zonepool/zones-c1t1d0s0@zfsBE     262K      -   633M  -
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   - zfszone          installed  /zonepool/zones                native   shared
```

Ejemplo 5–7 Migración de un sistema de archivos raíz UFS con raíz de zona a un sistema de archivos raíz ZFS

En este ejemplo, un sistema Oracle Solaris 10 9/10 con un sistema de archivos raíz UFS y una raíz de zona (`/uzone/ufszone`), así como una agrupación que no es raíz ZFS (`pool`) y una raíz de zona (`/pool/zfszone`) se migra a un sistema de archivos raíz ZFS. Asegúrese de que la agrupación raíz ZFS se haya creado y de que las zonas se hayan instalado e iniciado antes de intentar la migración.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 ufszone          running    /uzone/ufszone                 native   shared
   3 zfszone          running    /pool/zones/zfszone            native   shared
```

```
# lucreate -c ufsBE -n zfsBE -p rpool
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c1t1d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Populating file systems on boot environment <zfsBE>.
Checking selection integrity.
Integrity check OK.
```

```
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Copying root of zone <ufszone> to </.alt.tmp.b-EYd.mnt/uzone/ufszone>.
Creating snapshot for <pool/zones/zfszone> on <pool/zones/zfszone@zfsBE>.
Creating clone for <pool/zones/zfszone@zfsBE> on <pool/zones/zfszone-zfsBE>.
Creating compare databases for boot environment <zfsBE>.
Creating compare database for file system </rpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfsBE>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-DLd.mnt
updating /.alt.tmp.b-DLd.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
ufsBE                      yes      yes    yes       no     -
zfsBE                      yes      no     no        yes    -
# luactivate zfsBE
.
.
.
# init 6
.
.
.
# zfs list
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
pool                                 628M  66.3G    19K  /pool
pool/zones                           628M  66.3G    20K  /pool/zones
pool/zones/zfszone                  75.5K  66.3G   627M  /pool/zones/zfszone
pool/zones/zfszone-ufsBE             628M  66.3G   627M  /pool/zones/zfszone-ufsBE
pool/zones/zfszone-ufsBE@zfsBE        98K      -   627M  -
rpool                               7.76G  59.2G    95K  /rpool
rpool/ROOT                          5.25G  59.2G    18K  /rpool/ROOT
rpool/ROOT/zfsBE                    5.25G  59.2G  5.25G  /
rpool/dump                          2.00G  59.2G  2.00G  -
rpool/swap                           517M  59.7G    16K  -
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   - ufszone          installed  /uzone/ufszone                 native   shared
   - zfszone          installed  /pool/zones/zfszone            native   shared
```

Compatibilidad de ZFS con dispositivos de intercambio y volcado

Durante la instalación inicial de un sistema operativo Solaris o después de la ejecución de Actualización automática de Oracle Solaris desde un sistema de archivos UFS, se crea un área de intercambio en un volumen ZFS de la agrupación raíz ZFS. Por ejemplo:

```
# swap -l
swapfile                  dev  swaplo  blocks   free
/dev/zvol/dsk/rpool/swap 256,1      16 4194288 4194288
```

Durante la instalación inicial de un sistema operativo Solaris o la ejecución de Actualización automática de Oracle Solaris desde un sistema de archivos UFS, se crea un dispositivo de volcado en un volumen ZFS de la agrupación raíz ZFS. En general, un dispositivo de volcado no requiere administración porque se configura automáticamente en el momento de la instalación. Por ejemplo:

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
  Savecore enabled: yes
   Save compressed: on
```

Si inhabilita y elimina el dispositivo de volcado, tendrá que habilitarlo con el comando `dumpadm` después de que se haya vuelto a crear. En la mayoría de los casos, sólo tendrá que ajustar el tamaño del dispositivo de volcado mediante el comando `zfs`.

Para obtener información sobre el tamaño de los volúmenes de intercambio y volcado creados por los programas de instalación, consulte "Requisitos de instalación de Oracle Solaris y de Actualización automática de Oracle Solaris para compatibilidad con ZFS" en la página 127.

Tanto el tamaño del volumen de intercambio como el tamaño del volumen de volcado se pueden ajustar durante y después de la instalación. Para obtener más información, consulte "Ajuste del tamaño de los dispositivos de intercambio y volcado ZFS" en la página 168.

Al trabajar con dispositivos de intercambio y volcado ZFS, debe tener en cuenta los problemas siguientes:

- Para el área de intercambio y los dispositivos de volcado deben utilizarse volúmenes ZFS distintos.
- En la actualidad, no es posible utilizar un archivo de intercambio en un sistema de archivos ZFS.
- Si tiene que cambiar el área de intercambio o el dispositivo de volcado después de instalar o actualizar el sistema, utilice los comandos `swap` y `dumpadm` como en las versiones anteriores de Solaris. Si desea más información, consulte el Capítulo 20, "Configuring Additional Swap Space (Tasks)" de *System Administration Guide: Devices and File Systems* and Capítulo 17, "Managing System Crash Information (Tasks)" de *System Administration Guide: Advanced Administration*.

Consulte las secciones siguientes para obtener más información:

- "Ajuste del tamaño de los dispositivos de intercambio y volcado ZFS" en la página 168
- "Resolución de problemas de dispositivos de volcado ZFS" en la página 169

Ajuste del tamaño de los dispositivos de intercambio y volcado ZFS

Debido a las diferencias en la forma en que una instalación raíz ZFS determina el tamaño de los dispositivos de intercambio y volcado, podría ser que tuviera que ajustar el tamaño de dichos dispositivos antes, durante o después de la instalación.

- Durante una instalación inicial puede ajustar el tamaño de los volúmenes de intercambio y volcado. Para obtener más información, consulte el Ejemplo 5–1.
- Antes de ejecutar Actualización automática de Oracle Solaris puede crear y establecer el tamaño de los volúmenes de intercambio y volcado. Por ejemplo:

  1. Cree la agrupación de almacenamiento.

     ```
     # zpool create rpool mirror c0t0d0s0 c0t1d0s0
     ```

  2. Cree el dispositivo de volcado.

     ```
     # zfs create -V 2G rpool/dump
     ```

  3. Habilite el dispositivo de volcado.

     ```
     # dumpadm -d /dev/zvol/dsk/rpool/dump
           Dump content: kernel pages
            Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
     Savecore directory: /var/crash/t2000
       Savecore enabled: yes
        Save compressed: on
     ```

  4. Seleccione una de las opciones siguientes para crear el área de intercambio:

     - SPARC: cree su área de intercambio. Establezca el tamaño de bloque en 8 Kbytes.

       ```
       # zfs create -V 2G -b 8k rpool/swap
       ```

     - x86: cree su área de intercambio. Establezca el tamaño de bloque en 4 Kbytes.

       ```
       # zfs create -V 2G -b 4k rpool/swap
       ```

  5. Se debe habilitar el área de intercambio cuando se agrega o cambia un nuevo dispositivo de intercambio.
  6. Agregue una entrada para el volumen de intercambio en el archivo `/etc/vfstab`.

  Actualización automática de Oracle Solaris no cambia el tamaño de volúmenes de intercambio y volcado ya establecidos.

- Puede volver a configurar la propiedad `volsize` del dispositivo de volcado tras haber instalado un sistema. Por ejemplo:

  ```
  # zfs set volsize=2G rpool/dump
  # zfs get volsize rpool/dump
  NAME         PROPERTY  VALUE       SOURCE
  rpool/dump   volsize   2G          -
  ```

- Puede cambiar el tamaño del volumen de intercambio pero hasta que CR 6765386 esté integrado, es mejor quitar el dispositivo de intercambio en primer lugar. A continuación, vuelva a crearlo. Por ejemplo:

  ```
  # swap -d /dev/zvol/dsk/rpool/swap
  # zfs volsize=2G rpool/swap
  # swap -a /dev/zvol/dsk/rpool/swap
  ```

  Para obtener información sobre cómo quitar un dispositivo de intercambio en un sistema activo, consulte este sitio:

  http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide

- Puede ajustar el tamaño de los volúmenes de intercambio y volcado de un perfil de JumpStart mediante una sintaxis de perfil similar a la siguiente:

  ```
  install_type initial_install
  cluster SUNWCXall
  pool rpool 16g 2g 2g c0t0d0s0
  ```

  En este perfil, dos entradas `2g` establecen el tamaño del volumen de intercambio y volcado en 2 GB cada uno.

- Si necesita más espacio de intercambio en un sistema ya instalado, simplemente agregue otro volumen de intercambio. Por ejemplo:

  ```
  # zfs create -V 2G rpool/swap2
  ```

  A continuación, active el nuevo volumen de intercambio. Por ejemplo:

  ```
  # swap -a /dev/zvol/dsk/rpool/swap2
  # swap -l
  swapfile                  dev  swaplo   blocks   free
  /dev/zvol/dsk/rpool/swap  256,1      16 1058800 1058800
  /dev/zvol/dsk/rpool/swap2 256,3      16 4194288 4194288
  ```

  Por último, agregue una entrada para el segundo volumen de intercambio en el archivo `/etc/vfstab` .

Resolución de problemas de dispositivos de volcado ZFS

Revise los siguientes elementos si tiene problemas al capturar un volcado de bloqueo del sistema o al cambiar el tamaño del dispositivo de volcado.

- Si no se creó automáticamente un volcado de bloqueo, puede utilizar el comando `savecore` para guardar el volcado de bloqueo.
- Un volumen de volcado se crea automáticamente cuando se instala inicialmente un sistema de archivos raíz ZFS o se migra a un sistema de archivos ZFS. En la mayoría de los casos, sólo será necesario ajustar el tamaño del volumen de volcado si el tamaño del volumen de

volcado predeterminado es demasiado pequeño. Por ejemplo, en un sistema de mucha memoria, el tamaño del volumen de volcado se aumenta a 40 GB como sigue:

```
# zfs set volsize=40G rpool/dump
```

El cambio de tamaño de un volumen de volcado puede ser un proceso largo.

Si, por cualquier razón, tiene que habilitar un dispositivo de volcado tras crear un dispositivo de volcado manualmente, utilice sintaxis similar a la siguiente:

```
# dumpadm -d /dev/zvol/dsk/rpool/dump
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
  Savecore enabled: yes
```

- Un sistema con una memoria de 128 GB o mayor necesitará un dispositivo de volcado mayor que el dispositivo de volcado que se crea de forma predeterminada. Si el dispositivo de volcado es demasiado pequeño para capturar un volcado de bloqueo existente, se muestra un mensaje parecido al siguiente:

  ```
  # dumpadm -d /dev/zvol/dsk/rpool/dump
  dumpadm: dump device /dev/zvol/dsk/rpool/dump is too small to hold a system dump
  dump size 36255432704 bytes, device size 34359738368 bytes
  ```

  Para obtener información sobre el cambio de tamaño de los dispositivos de intercambio y volcado, consulte "Planning for Swap Space" de *System Administration Guide: Devices and File Systems*.

- No se puede agregar actualmente un dispositivo de volcado a una agrupación con varios dispositivos de nivel superior. Verá un mensaje similar al siguiente:

```
# dumpadm -d /dev/zvol/dsk/datapool/dump
dump is not supported on device '/dev/zvol/dsk/datapool/dump': 'datapool' has multiple top level vdevs
```

  Agregue el dispositivo de volcado a la agrupación raíz, que no puede tener varios dispositivos de nivel superior.

Inicio desde un sistema de archivos raíz ZFS

Tanto los sistemas basados en SPARC como en x86 utilizan el nuevo estilo de inicio con un archivo de almacenamiento de inicio, que consiste en una imagen de sistema de archivos con los archivos que se necesitan para iniciar. Si se inicia un sistema desde un sistema de archivos raíz ZFS, los nombres de ruta del archivo de almacenamiento de inicio y del archivo de núcleo se resuelven en el sistema de archivos raíz que se selecciona para iniciar.

Cuando se inicia un sistema para la instalación, se usa un disco RAM para el sistema de archivos raíz durante todo el proceso de instalación.

El inicio desde un sistema de archivos ZFS es diferente de un sistema de archivos UFS porque, con ZFS, el especificador de dispositivos de inicio identifica una agrupación de almacenamiento, no un solo sistema de archivos raíz. Una agrupación de almacenamiento puede contener varios *conjuntos de datos que se pueden iniciar* o sistemas de archivos raíz ZFS. Si se inicia desde ZFS, debe especificar un dispositivo de inicio y un sistema de archivos raíz en la agrupación identificada por el dispositivo de inicio.

De forma predeterminada, el conjunto de datos seleccionado para iniciar es el que queda identificado por la propiedad `bootfs` de la agrupación. Esta selección predeterminada se puede modificar optando por un conjunto de datos alternativo que se puede iniciar en el comando `boot -Z`.

Inicio desde un disco alternativo en una agrupación raíz ZFS reflejada

Puede crear una agrupación raíz ZFS reflejada al instalar el sistema; también puede conectar un disco para crear una agrupación raíz ZFS reflejada tras la instalación. Para más información, consulte:

- "Instalación de un sistema de archivos raíz ZFS (instalación inicial)" en la página 130
- "Cómo crear una agrupación raíz reflejada (posterior a la instalación)" en la página 136

Revise los siguientes problemas conocidos relativos a agrupaciones raíz ZFS reflejadas:

- CR 6668666: debe instalar la información de inicio en los discos conectados adicionalmente mediante los comandos `installboot` o `installgrub` si desea habilitar el inicio en los otros discos del reflejo. Si crea una agrupación raíz ZFS reflejada con el método de instalación inicial, este paso no es necesario. Por ejemplo, si `c0t1d0s0` era el segundo disco agregado al reflejo, la sintaxis del comando `installboot` o `installgrub` sería la siguiente:
  - SPARC:

```
sparc# installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk /dev/rdsk/c0t1d0s0
```

  - x86:

```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c0t1d0s0
```

- Puede iniciar desde distintos dispositivos en una agrupación raíz ZFS reflejada. Según la configuración de hardware, quizá deba actualizar la PROM o el BIOS para especificar otro dispositivo de inicio.

  Por ejemplo, puede iniciar desde cualquier disco (`c1t0d0s0` o `c1t1d0s0`) de la siguiente agrupación.

```
# zpool status
  pool: rpool
 state: ONLINE
 scrub: none requested
```

```
config:

        NAME          STATE     READ WRITE CKSUM
        rpool         ONLINE       0     0     0
          mirror-0    ONLINE       0     0     0
            c1t0d0s0  ONLINE       0     0     0
            c1t1d0s0  ONLINE       0     0     0
```

- SPARC: introduzca el disco alternativo en el indicador `Aceptar`.

  ```
  ok boot /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
  ```

  Tras reiniciar el sistema, confirme el dispositivo de inicio activo. Por ejemplo:

  ```
  SPARC# prtconf -vp | grep bootpath
         bootpath:  '/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a'
  ```

- x86: seleccione un disco alternativo en la agrupación raíz ZFS reflejada en el menú del BIOS pertinente.

  A continuación, use una sintaxis similar a la siguiente para confirmar que ha iniciado desde el disco alternativo:

  ```
  x86# prtconf -v|sed -n '/bootpath/,/value/p'
         name='bootpath' type=string items=1
             value='/pci@0,0/pci8086,25f8@4/pci108e,286@0/disk@0,0:a'
  ```

SPARC: inicio desde un sistema de archivos raíz ZFS

En un sistema basado en SPARC con varios entornos de inicio ZFS, puede iniciar desde cualquier entorno de inicio mediante el comando `luactivate`.

Durante el proceso de instalación del sistema operativo Solaris y la ejecución de Actualización automática de Oracle Solaris, el sistema de archivos raíz ZFS se designa automáticamente mediante la propiedad `bootfs`.

En una agrupación puede haber varios conjuntos de datos que se pueden iniciar. De forma predeterminada, la entrada del conjunto de datos que se puede iniciar del archivo */nombre_agrupación*`/boot/menu.lst` se identifica mediante la propiedad `bootfs` de la agrupación. Ahora bien, una entrada de `menu.lst`puede contener un comando `bootfs`, que especifica un conjunto de datos alternativo de la agrupación. De esta manera, el archivo `menu.lst` puede contener entradas de varios sistemas de archivos raíz dentro de la agrupación.

Si un sistema se instala con un sistema de archivos raíz ZFS o se migra a un sistema de archivos raíz ZFS, al archivo `menu.lst` se le agrega una entrada similar a la siguiente:

```
title zfsBE
bootfs rpool/ROOT/zfsBE
title zfs2BE
bootfs rpool/ROOT/zfs2BE
```

Al crearse un entorno de inicio, se actualiza automáticamente el archivo `menu.lst`.

En un sistema basado en SPARC hay dos nuevas opciones de inicio:

- Después de activar el entorno de inicio, puede utilizar el comando de inicio `-L` para obtener una lista de conjuntos de datos que se pueden iniciar en una agrupación ZFS. A continuación, puede seleccionar en la lista uno de los conjuntos de datos que se pueden iniciar. Se muestran instrucciones pormenorizadas para iniciar dicho conjunto de datos. El conjunto de datos seleccionado se puede iniciar siguiendo esas instrucciones.
- Puede utilizar el comando de inicio `-Z` *conjunto_datos* para iniciar un determinado conjunto de datos ZFS.

EJEMPLO 5–8 SPARC: inicio desde un determinado entorno de inicio ZFS

Si dispone de varios entornos de inicio ZFS en una agrupación de almacenamiento ZFS en el dispositivo de inicio del sistema, puede utilizar el comando `luactivate` para designar un entorno de inicio predeterminado.

Por ejemplo, los siguientes entornos de inicio ZFS están disponibles como se describe en la salida de `lustatus`:

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      no     no        yes    -
zfs2BE                     yes      yes    yes       no     -
```

Si tiene varios entornos de inicio ZFS en un sistema basado en SPARC, puede utilizar el comando `boot -L` para iniciar desde un entorno de inicio que sea diferente del predeterminado. Sin embargo, un entorno de inicio que se inicia desde una sesión `boot -L` no se restablece como el predeterminado, ni se actualiza la propiedad `bootfs`. Si desea que el entorno de inicio que se inicia desde una sesión `boot -L` sea el predeterminado, debe activarlo con el comando `luactivate`.

Por ejemplo:

```
ok boot -L
Rebooting with command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0  File and args: -L

1 zfsBE
2 zfs2BE
Select environment to boot: [ 1 - 2 ]: 1
To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfsBE

Program terminated
ok boot -Z rpool/ROOT/zfsBE
```

EJEMPLO 5–9 SPARC: inicio de un sistema de archivos ZFS en modo a prueba de fallos

En un sistema basado en SPARC, puede iniciar desde el archivo de almacenamiento a prueba de fallos ubicado en `/platform/‘uname -i‘/failsafe` como se muestra a continuación:

```
ok boot -F failsafe
```

Para iniciar un archivo de almacenamiento a prueba de fallos desde un determinado conjunto de datos ZFS que se puede iniciar, utilice una sintaxis similar a la siguiente:

```
ok boot -Z rpool/ROOT/zfsBE -F failsafe
```

x86: inicio desde un sistema de archivos raíz ZFS

Las entradas siguientes se agregan al archivo */pool-name* `/boot/grub/menu.lst` durante el proceso de instalación o al ejecutarse Actualización automática de Oracle Solaris para iniciar ZFS de forma automática:

```
title Solaris 10 9/10  X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Si el dispositivo que GRUB identifica como dispositivo de inicio contiene una agrupación de almacenamiento ZFS, el archivo `menu.lst` se utiliza para crear el menú GRUB.

En el caso de un sistema basado en x86 con varios entornos de inicio ZFS, el entorno de inicio se puede seleccionar en el menú GRUB. Si el sistema de archivos raíz correspondiente a esta entrada de menú es un conjunto de datos ZFS, se agrega la opción siguiente:

```
-B $ZFS-BOOTFS
```

EJEMPLO 5–10 x86: inicio de un sistema de archivos ZFS

Si se inicia un sistema desde un sistema de archivos ZFS, el parámetro `-B $ZFS-BOOTFS` especifica el dispositivo raíz en la línea `kernel` o `module` en la entrada del menú GRUB. El GRUB pasa al núcleo el valor de este parámetro, parecido a todos los parámetros que especifica la opción `-B`. Por ejemplo:

```
title Solaris 10 9/10  X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
```

```
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

EJEMPLO 5–11 x86: inicio de un sistema de archivos ZFS en modo a prueba de fallos

El archivo de almacenamiento a prueba de fallos de x86 es `/boot/x86.miniroot-safe` y se puede iniciar seleccionando la entrada a prueba de fallos de Solaris en el menú GRUB. Por ejemplo:

```
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Resolución de problemas de punto de montaje ZFS que impiden un inicio correcto (Solaris 10 10/08)

El uso del comando `luactivate` es la mejor forma de cambiar el entorno de inicio activo. Si el entorno de inicio activo no puede iniciarse, debido a un parche incorrecto o a un error de configuración, la única forma de iniciar otro entorno de inicio es seleccionar dicho entorno en el momento de iniciar. Puede seleccionar un entorno de inicio alternativo en el menú GRUB en un sistema basado en x86, o iniciarlo explícitamente desde la PROM de un sistema basado en SPARC.

Debido a un error en Actualización automática de Oracle Solaris en la versión Solaris 10 10/08, el inicio del entorno de inicio no activo podría fallar porque un conjunto de datos ZFS o el conjunto de datos ZFS de una zona del entorno de inicio tiene un punto de montaje no válido. Ese mismo error impide el montaje del entorno de inicio si tiene un conjunto de datos `/var` aparte.

Si el conjunto de datos de una zona tiene un punto de montaje no válido, éste se puede corregir si se realizan los siguientes pasos.

▼ Cómo resolver problemas de punto de montaje ZFS

1. **Inicie el sistema desde un archivo de almacenamiento a prueba de fallos.**

2. **Importe la agrupación.**

   Por ejemplo:

   ```
   # zpool import rpool
   ```

3 **Busque puntos de montaje temporales incorrectos.**

Por ejemplo:

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10u6

    NAME                                MOUNTPOINT
    rpool/ROOT/s10u6                    /.alt.tmp.b-VP.mnt/
    rpool/ROOT/s10u6/zones              /.alt.tmp.b-VP.mnt//zones
    rpool/ROOT/s10u6/zones/zonerootA    /.alt.tmp.b-VP.mnt/zones/zonerootA
```

El punto de montaje del entorno de inicio raíz (`rpool/ROOT/s10u6`) debe ser `/`.

Si falla el inicio debido a problemas de montaje de `/var`, busque un punto de montaje temporal similar incorrecto para el conjunto de datos `/var`.

4 **Restablezca los puntos de montaje del entorno de inicio ZFS y sus conjuntos de datos.**

Por ejemplo:

```
# zfs inherit -r mountpoint rpool/ROOT/s10u6
# zfs set mountpoint=/ rpool/ROOT/s10u6
```

5 **Reinicie el sistema.**

Si se presenta la opción para iniciar un determinado entorno de inicio, en el menú GRUB o el indicador de OpenBoot PROM, seleccione el entorno de inicio cuyos puntos de montaje se han acabado de corregir.

Inicio en situaciones de recuperación en un entorno raíz ZFS

Utilice el procedimiento siguiente si necesita iniciar el sistema para recuperarse de la pérdida de una contraseña raíz o de un problema similar.

Dependiendo de la gravedad del error, tendrá que iniciar en modo a prueba de fallos o desde un medio alternativo. En general, puede iniciar en modo a prueba de fallos para recuperar una contraseña raíz perdida o desconocida.

- “Cómo iniciar ZFS en modo a prueba de fallos” en la página 177
- “Cómo iniciar ZFS desde un medio alternativo” en la página 177

Si necesita recuperar una agrupación raíz o una instantánea de agrupación raíz, consulte “Recuperación de la agrupación raíz ZFS o las instantáneas de la agrupación raíz” en la página 178.

▼ Cómo iniciar ZFS en modo a prueba de fallos

1 Inicie en modo a prueba de fallos.

En un sistema SPARC:

```
ok boot -F failsafe
```

En un sistema x86, seleccione el modo a prueba de fallos en el indicador de GRUB.

2 Monte el entorno de inicio ZFS en `/a` cuando se le solicite:

```
.
.
.
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
Starting shell.
```

3 Cambie al directorio `/a/etc`.

```
# cd /a/etc
```

4 Si es necesario, establezca el tipo TERM.

```
# TERM=vt100
# export TERM
```

5 Corrija el archivo `passwd` o `shadow`.

```
# vi shadow
```

6 Reinicie el sistema.

```
# init 6
```

▼ Cómo iniciar ZFS desde un medio alternativo

Si un problema impide que el sistema se inicie correctamente, o se produce algún otro problema grave, deberá iniciar desde un servidor de instalación en red o desde un CD de instalación de Solaris, importar la agrupación raíz, montar el entorno de inicio ZFS e intentar resolver el problema.

1 Inicie desde un CD de instalación o desde la red.

- SPARC:

  ```
  ok boot cdrom -s
  ok boot net -s
  ```

  Si no utiliza la opción `-s`, deberá salir del programa de instalación.

- x86: seleccione la opción de inicio de red o de inicio desde el CD local.

2 **Importe la agrupación raíz y especifique un punto de montaje alternativo. Por ejemplo:**

```
# zpool import -R /a rpool
```

3 **Monte el entorno de inicio ZFS. Por ejemplo:**

```
# zfs mount rpool/ROOT/zfsBE
```

4 **Acceda al contenido ZFS desde el directorio `/a`.**

```
# cd /a
```

5 **Reinicie el sistema.**

```
# init 6
```

Recuperación de la agrupación raíz ZFS o las instantáneas de la agrupación raíz

Las siguientes secciones describen cómo realizar las siguientes tareas:

- "Cómo sustituir un disco en la agrupación raíz ZFS" en la página 178
- "Cómo crear instantáneas de la agrupación raíz" en la página 180
- "Cómo volver a crear una agrupación raíz ZFS y restaurar instantáneas de agrupaciones raíz" en la página 182
- "Cómo deshacer instantáneas de agrupaciones raíz a partir de un inicio a prueba de fallos" en la página 183

▼ Cómo sustituir un disco en la agrupación raíz ZFS

Es posible que necesite sustituir un disco en la agrupación raíz, por los siguientes motivos:

- La agrupación raíz es demasiado pequeña y desea sustituir un disco pequeño por uno mayor.
- Un disco de la agrupación raíz no funciona correctamente. En una agrupación no redundante, si el disco falla y el sistema no se inicia, deberá iniciar desde un medio alternativo, como un CD o la red, antes de sustituir el disco de la agrupación raíz.

En una configuración de agrupación raíz reflejada, puede intentar una sustitución de discos sin iniciar desde un soporte alternativo. Puede sustituir un disco averiado mediante el comando `zpool replace`. O, si tiene un disco adicional, puede utilizar el comando `zpool attach`. Consulte el procedimiento de esta sección para ver un ejemplo de cómo conectar un disco adicional y la desconexión de un disco de agrupación raíz.

Algunos dispositivos de hardware requieren que se desconecte un disco y se desconfigure antes de intentar la operación `zpool replace` para sustituir un disco averiado. Por ejemplo:

```
# zpool offline rpool c1t0d0s0
# cfgadm -c unconfigure c1::dsk/c1t0d0
<Physically remove failed disk c1t0d0>
<Physically insert replacement disk c1t0d0>
# cfgadm -c configure c1::dsk/c1t0d0
# zpool replace rpool c1t0d0s0
# zpool online rpool c1t0d0s0
# zpool status rpool
<Let disk resilver before installing the boot blocks>
SPARC# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdsk/c1t0d0s0
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1t9d0s0
```

En algunos dispositivos de hardware, no es necesario que el disco de sustitución esté conectado ni reconfigurarlo después de insertarlo.

Debe identificar los nombres de ruta del dispositivo de inicio de los discos nuevo y actual para poder probar el inicio desde el disco de sustitución y también iniciar manualmente desde el disco existente, si el disco de sustitución falla. En el ejemplo que aparece en el siguiente procedimiento, el nombre de la ruta del disco de agrupación raíz actual es (`c1t10d0s0`):

```
/pci@8,700000/pci@3/scsi@5/sd@a,0
```

El nombre de ruta del disco de inicio de sustitución es (`c1t9d0s0`):

```
/pci@8,700000/pci@3/scsi@5/sd@9,0
```

1. **Conecte físicamente el disco de sustitución (o nuevo).**

2. **Confirme que el disco nuevo tiene una etiqueta SMI y un segmento 0.**

   Para obtener información sobre el reetiquetado de un disco que está diseñado para la agrupación raíz, consulte el sitio siguiente:

   `http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide`

3. **Conecte el nuevo disco a la agrupación raíz.**

   Por ejemplo:

   ```
   # zpool attach rpool c1t10d0s0 c1t9d0s0
   ```

4. **Confirme el estado de la agrupación raíz.**

   Por ejemplo:

   ```
   # zpool status rpool
     pool: rpool
    state: ONLINE
   status: One or more devices is currently being resilvered.  The pool will
           continue to function, possibly in a degraded state.
   action: Wait for the resilver to complete.
    scrub: resilver in progress, 25.47% done, 0h4m to go
   config:
   ```

```
        NAME           STATE     READ WRITE CKSUM
        rpool          ONLINE       0     0     0
          mirror-0     ONLINE       0     0     0
            c1t10d0s0  ONLINE       0     0     0
            c1t9d0s0   ONLINE       0     0     0

errors: No known data errors
```

5 Cuando se haya completado la creación, aplique los bloques de inicio al nuevo disco.

Utilizando una sintaxis similar a la siguiente:

- SPARC:

  ```
  # installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk /dev/rdsk/c1t9d0s0
  ```

- x86:

  ```
  # installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1t9d0s0
  ```

6 Compruebe que puede iniciar desde el nuevo disco.

Por ejemplo, en un sistema basado en SPARC, deberá usar una sintaxis similar a la siguiente:

```
ok boot /pci@8,700000/pci@3/scsi@5/sd@9,0
```

7 Si el sistema se inicia desde el nuevo disco, desconecte el disco antiguo.

Por ejemplo:

```
# zpool detach rpool c1t10d0s0
```

8 Configure el sistema para que se inicie automáticamente desde el nuevo disco, ya sea mediante el comando `eeprom`, el comando `setenv` desde la PROM de inicio de SPARC, o bien vuelva a configurar el BIOS del equipo.

▼ Cómo crear instantáneas de la agrupación raíz

Puede crear instantáneas de la agrupación raíz para las recuperaciones. La forma más recomendable de crear instantáneas de agrupaciones raíz es realizar una instantánea recursiva de la agrupación raíz.

El procedimiento siguiente crea una instantánea de agrupación raíz recursiva y almacena la instantánea como un archivo en una agrupación en un sistema remoto. Si una agrupación raíz falla, el conjunto de datos remoto se puede montar mediante NFS y el archivo de instantánea se puede recibir en la agrupación que se ha vuelto a crear. O bien puede almacenar instantáneas de agrupaciones raíz como las instantáneas reales en una agrupación de un sistema remoto. Enviar y recibir las instantáneas desde un sistema remoto es un poco más complicado porque se debe configurar `ssh` o utilizar `rsh` mientras el sistema que hay que reparar se inicia desde la miniraíz del sistema operativo Solaris.

Para obtener información sobre el almacenamiento y la recuperación de forma remota de instantáneas de agrupación raíz, así como la información más actualizada sobre la recuperación de agrupaciones raíz, vaya a este sitio:

http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide

La validación de instantáneas almacenadas remotamente como archivos o instantáneas es un paso importante en la recuperación de una agrupación raíz. Con cualquiera de estos métodos, las instantáneas se deben volver a crear de forma rutinaria, como, por ejemplo, cuando la configuración de la agrupación cambia o cuando se actualiza el sistema operativo Solaris.

En el procedimiento siguiente, el sistema se inicia desde el entorno de inicio `zfsBE`.

1 Cree una agrupación y un sistema de archivos en un sistema remoto para almacenar las instantáneas.

Por ejemplo:

```
remote# zfs create rpool/snaps
```

2 Comparta el sistema de archivos con el sistema local.

Por ejemplo:

```
remote# zfs set sharenfs='rw=local-system,root=local-system' rpool/snaps
# share
-@rpool/snaps   /rpool/snaps   sec=sys,rw=local-system,root=local-system   ""
```

3 Cree una instantánea recursiva de la agrupación raíz.

```
local# zfs snapshot -r rpool@0804
local# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
rpool                     6.17G  60.8G    98K  /rpool
rpool@0804                   0      -    98K  -
rpool/ROOT                4.67G  60.8G    21K  /rpool/ROOT
rpool/ROOT@0804              0      -    21K  -
rpool/ROOT/zfsBE          4.67G  60.8G  4.67G  /
rpool/ROOT/zfsBE@0804      386K      -  4.67G  -
rpool/dump                1.00G  60.8G  1.00G  -
rpool/dump@0804              0      -  1.00G  -
rpool/swap                 517M  61.3G    16K  -
rpool/swap@0804              0      -    16K  -
```

4 Envíe las instantáneas de la agrupación raíz al sistema remoto.

Por ejemplo:

```
local# zfs send -Rv rpool@0804 > /net/remote-system/rpool/snaps/rpool.0804
sending from @ to rpool@0804
sending from @ to rpool/swap@0804
sending from @ to rpool/ROOT@0804
sending from @ to rpool/ROOT/zfsBE@0804
sending from @ to rpool/dump@0804
```

▼ Cómo volver a crear una agrupación raíz ZFS y restaurar instantáneas de agrupaciones raíz

En este procedimiento, suponga las siguientes condiciones:

- La agrupación raíz ZFS no se puede recuperar.
- Las instantáneas de las agrupaciones raíz ZFS se almacenan en un sistema remoto y se comparten a través de NFS.

Todos los pasos se llevan a cabo en el sistema local.

1 Efectúe el inicio desde el CD/DVD o desde la red.

- SPARC: seleccione uno de los siguientes métodos de inicio:

```
ok boot net -s
ok boot cdrom -s
```

  Si no utiliza la opción `-s`, deberá salir del programa de instalación.

- x86: seleccione la opción para iniciar desde el DVD o desde la red. A continuación, salga del programa de instalación.

2 Monte el conjunto de datos remoto de instantáneas.

Por ejemplo:

```
# mount -F nfs remote-system:/rpool/snaps /mnt
```

Si los servicios de red no están configurados, es posible que deba especificar la dirección IP del *sistema remoto*.

3 Si se reemplaza el disco de la agrupación raíz y no contiene una etiqueta de disco que sea utilizable por ZFS, deberá etiquetar de nuevo el disco.

Para obtener más información sobre cómo volver a etiquetar el disco, consulte el sitio siguiente:

`http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide`

4 Vuelva a crear la agrupación raíz.

Por ejemplo:

```
# zpool create -f -o failmode=continue -R /a -m legacy -o cachefile=
/etc/zfs/zpool.cache rpool c1t1d0s0
```

5 Restaure las instantáneas de agrupaciones raíz.

Este paso puede tardar algo. Por ejemplo:

```
# cat /mnt/rpool.0804 | zfs receive -Fdu rpool
```

El uso de la opción -u significa que el archivo de almacenamiento restaurado no está montado cuando se completa la operación `zfs receive`.

6 Compruebe que los conjuntos de datos de agrupaciones raíz se hayan restaurado.

Por ejemplo:

```
# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
rpool                     6.17G  60.8G    98K  /a/rpool
rpool@0804                   0      -    98K  -
rpool/ROOT                4.67G  60.8G    21K  /legacy
rpool/ROOT@0804              0      -    21K  -
rpool/ROOT/zfsBE          4.67G  60.8G  4.67G  /a
rpool/ROOT/zfsBE@0804      398K      -  4.67G  -
rpool/dump                1.00G  60.8G  1.00G  -
rpool/dump@0804              0      -  1.00G  -
rpool/swap                 517M  61.3G    16K  -
rpool/swap@0804              0      -    16K  -
```

7 Defina la propiedad `bootfs` en el entorno de inicio de la agrupación raíz.

Por ejemplo:

```
# zpool set bootfs=rpool/ROOT/zfsBE rpool
```

8 Instale los bloques de inicio en el nuevo disco.

SPARC:

```
# installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk /dev/rdsk/c1t1d0s0
```

x86:

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1t1d0s0
```

9 Reinicie el sistema.

```
# init 6
```

▼ Cómo deshacer instantáneas de agrupaciones raíz a partir de un inicio a prueba de fallos

Este procedimiento da por hecho que las instantáneas de agrupaciones raíz existentes están disponibles. En el ejemplo, están disponibles en el sistema local.

```
# zfs snapshot -r rpool@0804
# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
rpool                     6.17G  60.8G    98K  /rpool
rpool@0804                   0      -    98K  -
rpool/ROOT                4.67G  60.8G    21K  /rpool/ROOT
rpool/ROOT@0804              0      -    21K  -
```

```
rpool/ROOT/zfsBE           4.67G  60.8G  4.67G  /
rpool/ROOT/zfsBE@0804       398K      -  4.67G  -
rpool/dump                 1.00G  60.8G  1.00G  -
rpool/dump@0804               0      -  1.00G  -
rpool/swap                  517M  61.3G    16K  -
rpool/swap@0804               0      -    16K  -
```

1 Apague el sistema e inicie en modo de inicio a prueba de fallos.

```
ok boot -F failsafe
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a

Starting shell.
```

2 Deshaga cada instantánea de agrupación raíz.

```
# zfs rollback rpool@0804
# zfs rollback rpool/ROOT@0804
# zfs rollback rpool/ROOT/zfsBE@0804
```

3 Vuelva a iniciar en modo multiusuario.

```
# init 6
```

◆ ◆ ◆ CAPÍTULO 6

Administrar sistemas de archivos ZFS de Oracle Solaris

Este capítulo ofrece información detallada sobre la administración de sistemas de archivos ZFS de Oracle Solaris. En este capítulo se incluyen conceptos como la disposición jerárquica del sistema de archivos, la herencia de propiedades, así como la administración automática de puntos de montaje y cómo compartir interacciones.

Este capítulo se divide en las secciones siguientes:

Administración de sistemas de archivos AFS (descripción general)

Un sistema de archivos ZFS se genera encima de una agrupación de almacenamiento. Los sistemas de archivos se pueden crear y destruir dinámicamente sin necesidad de asignar ni dar formato a ningún espacio en el disco subyacente. Debido a que los sistemas de archivos son tan ligeros y a que son el punto central de administración en ZFS, puede crear muchos de ellos.

Los sistemas de archivos ZFS se administran mediante el comando `zfs`. El comando `zfs` ofrece un conjunto de subcomandos que ejecutan operaciones específicas en los sistemas de archivos. Este capítulo describe estos subcomandos detalladamente. Las instantáneas, los volúmenes y los clones también se administran mediante este comando, pero estas funciones sólo se explican brevemente en este capítulo. Para obtener información detallada sobre instantáneas y clones, consulte el Capítulo 7, "Uso de clones e instantáneas de Oracle Solaris ZFS". Para obtener información detallada sobre volúmenes ZFS, consulte "Volúmenes de ZFS" en la página 281.

Nota – El término *conjunto de datos* se utiliza en este capítulo como término genérico para referirse a un sistema de archivos, instantánea, clon o volumen.

Creación, destrucción y cambio de nombre de sistemas de archivos ZFS

Los sistemas de archivos ZFS se pueden crear y destruir mediante los comandos `zfs create` y `zfs destroy`, respectivamente. Mediante el comando `zfs rename` se puede cambiar el nombre a los sistemas de archivos ZFS.

Creación de un sistema de archivos ZFS

Los sistemas de archivos ZFS se crean mediante el comando `zfs create`. El subcomando `create` toma un único argumento: el nombre del sistema de archivos que crear. El nombre del sistema de archivos se especifica como nombre de ruta que comienza por el nombre de la agrupación:

nombre_agrupación/[nombre_sistema_archivos/]nombre_sistema_archivos

El nombre de agrupación y los nombres del sistema de archivos inicial de la ruta identifican la ubicación en la jerarquía donde se creará el nuevo sistema de archivos. El último nombre de la ruta identifica el nombre del sistema de archivos que se creará. El nombre del sistema de archivos debe seguir las convenciones de denominación establecidas en "Requisitos de asignación de nombres de componentes de ZFS" en la página 51.

En el ejemplo siguiente, un sistema de archivos denominado `bonwick` se crea en el sistema de archivos `tank/home`.

```
# zfs create tank/home/bonwick
```

ZFS monta de forma automática el sistema de archivos recién creado si se crea correctamente. De forma predeterminada, los sistemas de archivos se montan como */conjunto de datos*, mediante la ruta proporcionada para el nombre del sistema de archivos en el subcomando `create`. En este ejemplo, el sistema de archivos recién creado `bonwick` se encuentra en `/tank/home/bonwick`. Para obtener más información sobre puntos de montaje que se administran automáticamente, consulte "Administración de puntos de montaje de ZFS" en la página 211.

Para obtener más información sobre el comando `zfs create`, consulte `zfs(1M)`.

Las propiedades del sistema de archivos pueden establecerse al crear dicho sistema de archivos.

En el ejemplo siguiente se crea un punto de montaje de `/export/zfs` para el sistema de archivos `tank/home`:

```
# zfs create -o mountpoint=/export/zfs tank/home
```

Para obtener más información sobre las propiedades del sistema de archivos, consulte "Introducción a las propiedades de ZFS" en la página 189.

Destrucción de un sistema de archivos ZFS

Para destruir un sistema de archivos ZFS, utilice el comando `zfs destroy`. El sistema de archivos destruido se desmonta automáticamente y se anula la compartición. Para obtener más información sobre puntos de montaje o recursos compartidos administrados automáticamente, consulte "Puntos de montaje automáticos" en la página 212.

En el ejemplo siguiente se destruye el sistema de archivos `tabriz`:

```
# zfs destroy tank/home/tabriz
```

Precaución – No aparece ningún mensaje de confirmación con el subcomando `destroy`. Utilícelo con extrema precaución.

Si el sistema de archivos que se desea destruir está ocupado y no se puede desmontar, el comando `zfs destroy` falla. Para destruir un sistema de archivos activo, utilice la opción `-f`. Úsela con precaución, puesto que puede desmontar, destruir y anular la compartición de sistemas de archivos activos, lo que provoca un comportamiento inesperado de la aplicación.

```
# zfs destroy tank/home/ahrens
cannot unmount 'tank/home/ahrens': Device busy

# zfs destroy -f tank/home/ahrens
```

El comando `zfs destroy` también falla si un sistema de archivos tiene descendientes. Para destruir repetidamente un sistema de archivos y todos sus descendientes, utilice la opción `-r`. Una destrucción repetitiva también destruye las instantáneas, por lo que debe utilizar esta opción con precaución.

```
# zfs destroy tank/ws
cannot destroy 'tank/ws': filesystem has children
use '-r' to destroy the following datasets:
tank/ws/billm
tank/ws/bonwick
tank/ws/maybee

# zfs destroy -r tank/ws
```

Si el sistema de archivos que se debe destruir tiene elementos dependientes indirectos, falla incluso el comando de destrucción repetitiva. Para forzar la destrucción de *todos* los dependientes, incluidos los sistemas de archivos clonados fuera de la jerarquía de destino, se debe utilizar la opción `-R`. Esta opción se debe utilizar con sumo cuidado.

```
# zfs destroy -r tank/home/schrock
cannot destroy 'tank/home/schrock': filesystem has dependent clones
use '-R' to destroy the following datasets:
tank/clones/schrock-clone

# zfs destroy -R tank/home/schrock
```

Precaución – No aparece ningún mensaje de confirmación con las opciones `-f`, `-r` o `-R` para el comando `zfs destroy`, por lo que debe utilizarlas con cuidado.

Para obtener información detallada sobre instantáneas y clones, consulte el Capítulo 7, "Uso de clones e instantáneas de Oracle Solaris ZFS".

Cambio de nombre de un sistema de archivos ZFS

Mediante el comando `zfs rename` se puede cambiar el nombre a los sistemas de archivos. Con el subcomando `rename` se pueden efectuar las operaciones siguientes:

- Cambiar el nombre de un sistema de archivos.
- Cambiar la ubicación del sistema de archivos en la jerarquía ZFS.
- Cambiar el nombre de un sistema de archivos y cambiar su ubicación dentro de la jerarquía ZFS.

El ejemplo siguiente utiliza el subcomando `renombrar` para cambiar de nombre de un sistema de archivos de `kustarz` a `kustarz_old` :

```
# zfs rename tank/home/kustarz tank/home/kustarz_old
```

El ejemplo siguiente muestra cómo utilizar `zfs rename` para cambiar la ubicación de un sistema de archivos:

```
# zfs rename tank/home/maybee tank/ws/maybee
```

En este ejemplo, el sistema de archivos `maybee` se reubica de `tank/home` a `tank/ws`. Si reubica un sistema de archivos mediante rename, la nueva ubicación debe estar en la misma agrupación y tener espacio suficiente en el disco para albergar este nuevo sistema de archivos. Si la nueva ubicación no tiene espacio suficiente en el disco, posiblemente por haber llegado a su cuota, fallará la operación `rename`.

Para obtener más información sobre las cuotas, consulte "Configuración de cuotas y reservas de ZFS" en la página 217.

La operación `rename` intenta una secuencia de desmontar/volver a montar para el sistema de archivos y los sistemas de archivos descendientes. El comando `rename` falla si la operación no puede desmontar un sistema de archivos activo. Si se produce este problema, deberá forzar el desmontaje del sistema de archivos.

Para obtener más información sobre el cambio de nombre de las instantáneas, consulte "Cambio de nombre de instantáneas de ZFS" en la página 228.

Introducción a las propiedades de ZFS

Las propiedades son para el mecanismo principal que utiliza para controlar el comportamiento de sistemas de archivos, volúmenes, instantáneas y clones. A menos que se indique otra cosa, las propiedades que se definen en la sección se aplican a todos los tipos de conjuntos de datos.

Las propiedades se dividen en dos tipos: nativas y definidas por el usuario. Las propiedades nativas exportan estadísticas internas o controlan el comportamiento del sistema de archivos ZFS. Asimismo, las propiedades nativas son configurables o de sólo lectura. Las propiedades del usuario no repercuten en el comportamiento del sistema de archivos ZFS, pero puede usarlas para anotar conjuntos de datos de forma que tengan sentido en su entorno. Para obtener más información sobre las propiedades del usuario, consulte "Propiedades de usuario de ZFS" en la página 202.

La mayoría de las propiedades configurables también se pueden heredar. Una propiedad que se puede heredar es aquella que, cuando se establece en un principal, se propaga a todos sus descendientes.

Todas las propiedades heredables tienen un origen asociado que indica la forma en que se ha obtenido una propiedad. El origen de una propiedad puede tener los valores siguientes:

`local`	Indica que la propiedad se ha establecido explícitamente en el conjunto de datos mediante el comando `zfs set`, tal como se describe en "Configuración de propiedades de ZFS" en la página 206.
`inherited from` *nombre_conjunto_datos*	Indica que la propiedad se ha heredado del superior nombrado.
`default`	Indica que el valor de la propiedad no se ha heredado o establecido localmente. Este origen es el resultado de que ningún superior tiene la propiedad como `local` de origen.

La tabla siguiente identifica las propiedades del sistema de archivos ZFS nativo configurable y de sólo lectura. Las propiedades nativas de sólo lectura se identifican como tales. Todas las demás propiedades nativas que se enumeran en esta tabla son configurables. Para obtener información sobre las propiedades del usuario, consulte "Propiedades de usuario de ZFS" en la página 202.

TABLA 6–1 Descripciones de propiedades nativas de ZFS

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`aclinherit`	Cadena	`secure`	Controla cómo se heredan las entradas de lista de control de acceso (ACL) cuando se crean los archivos y los directorios. Los valores son `discard`, `noallow`, `secure` y `passthrough`. Para obtener una descripción de estos valores, consulte "Propiedades de ACL" en la página 246.
`aclmode`	Cadena	`groupmask`	Controla cómo se modifica una entrada de lista de control de acceso (ACL) durante una operación de `chmod`. Los valores son `discard`, `groupmask` y `passthrough`. Para obtener una descripción de estos valores, consulte "Propiedades de ACL" en la página 246.
`atime`	Booleano	`on`	Controla si la hora de acceso de los archivos se actualiza cuando se leen. Si se desactiva esta propiedad, se evita la generación de tráfico de escritura al leer archivos y se puede mejorar considerablemente el rendimiento, si bien esto podría confundir a los programas de envío de correo y otras utilidades similares.
`available`	Número	N/D	Propiedad de sólo lectura que identifica la cantidad de espacio disponible para el conjunto de datos y todos los subordinados, suponiendo que no haya otra actividad en la agrupación. Como el espacio en el disco se comparte en una agrupación, el espacio disponible puede verse limitado por varios factores, como el tamaño físico de la agrupación, las cuotas, las reservas u otros conjuntos de datos de la agrupación. La abreviatura de la propiedad es `avail`. Para obtener más información sobre el cálculo de espacio, consulte "Cálculo del espacio de ZFS" en la página 62.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`canmount`	Booleano	`on`	Controla si un sistema de archivos determinado se puede montar con el comando `zfs mount`. Esta propiedad se puede establecer en cualquier sistema de archivos y la propiedad no es heredable. No obstante, cuando esta propiedad está establecida en `off`, los sistemas de archivos descendientes se pueden heredar, pero el sistema de archivos nunca se monta. Si se establece la opción `noauto`, un conjunto de datos sólo se puede montar y desmontar de manera explícita. El conjunto de datos no se monta automáticamente al crearse o importarse el conjunto de datos, ni se monta con el comando `zfs mount- a` ni se desmonta con el comando `zfs unmount-a`. Para obtener más información, consulte "Propiedad `canmount`" en la página 201.
`checksum`	Cadena	`on`	Controla la suma de comprobación utilizada para verificar la integridad de los datos. El valor predeterminado es `on`, que selecciona automáticamente un algoritmo adecuado, actualmente `fletcher4`. Los valores son `on, off, fletcher2` , `fletcher4` y `sha256`. El valor `off` deshabilita la comprobación de integridad en los datos del usuario. No se recomienda un valor de `off`.
`compression`	Cadena	`off`	Habilita o inhabilita la compresión de este conjunto de datos. Los valores son `on`, `off` y `lzjb`, `gzip` o `gzip-`*N*. En la actualidad, configurar esta propiedad en `lzjb`, `gzip` o `gzip-` *N* equivale a establecerla en `on`. Habilitar la compresión en un sistema de archivos en el que ya hay datos sólo comprime los datos nuevos. Los datos que existan están sin comprimir. La abreviatura de la propiedad es `compress`.
`compressratio`	Número	N/D	Propiedad de sólo lectura que identifica el índice de compresión alcanzado para un conjunto de datos, expresado como multiplicador. La compresión se puede activar ejecutando `zfs set compression=on` *conjunto_datos*. El valor se calcula a partir del tamaño lógico de todos los archivos y la cantidad de datos físicos a los que se hace referencia. Incluye grabaciones explícitas mediante el uso de la propiedad `compression`.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`copies`	Número	`1`	Establece la cantidad de copias de datos de usuarios por sistema de archivos. Los valores disponibles son `1`, `2`, o `3`. Estas copias son adicionales a cualquier redundancia de agrupación. El espacio en el disco que utilicen varias copias de datos de usuarios se carga en los pertinentes archivo y conjunto de datos, y se contabiliza en relación con las cuotas y reservas. Además, la propiedad `used` se actualiza si se habilitan varias copias. La configuración de esta propiedad debe considerarse al crear el sistema de archivos, puesto que, si se modifica la propiedad en cualquier sistema ya creado, sólo se afecta a los datos nuevos que se escriban.
`creation`	Cadena	N/D	Propiedad de sólo lectura que identifica la fecha y la hora de creación de este conjunto de datos.
`devices`	Booleano	`on`	Controla si se pueden abrir archivos de dispositivos en un sistema de archivos.
`exec`	Booleano	`on`	Controla si se permite ejecutar programas en un sistema de archivos. Asimismo, si se establece en `off`, no se permiten las llamadas de `mmap(2)` con `PROT_EXEC`.
`mounted`	Booleano	N/D	Propiedad de sólo lectura que indica si este sistema de archivos, un clon o una instantánea se encuentra montada. Esta propiedad no se aplica a los volúmenes. El valor puede ser `yes` o `no`.
`mountpoint`	Cadena	N/D	Controla el punto de montaje utilizado para este sistema de archivos. Si la propiedad `mountpoint` se cambia para un sistema de archivos, se desmontan éste y cualquier descendiente que herede el punto de montaje. Si el valor nuevo es `legacy`, permanecen desmontados. En cambio, se vuelven a montar automáticamente en la nueva ubicación si la propiedad era `legacy` o `none`, o bien si estaban montados antes de que cambiara la propiedad. Además, cualquier nuevo sistema de archivos compartidos se comparte o no en la ubicación nueva. Para obtener más información sobre el uso de esta propiedad, consulte "Administración de puntos de montaje de ZFS" en la página 211.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`primarycache`	Cadena	`all`	Controla la información que se guarda en la caché primaria (ARC). Los valores posibles son `all`, `none` y `metadata`. Si se establece en `all`, los datos de usuario y los metadatos se almacenan en la memoria caché. Si se establece en `none`, ni los datos de usuario ni los metadatos se almacenan en la memoria caché. Si se establece en `metadata`, sólo los metadatos se almacenan en la memoria caché.
`origin`	Cadena	N/D	Propiedad de sólo lectura para volúmenes o sistemas de archivos clónicos que identifica la instantánea a partir de la cual se ha creado el clon. No se puede destruir el origen (ni siquiera con las opciones `-r` o `-f`) en tanto exista un clon. Los sistemas de archivos no clónicos tienen la propiedad origin establecida en `none`.
`quota`	Número (o `none`)	`none`	Limita la cantidad de espacio en el disco que un conjunto de datos y sus descendientes pueden consumir. Esta propiedad fuerza un límite físico sobre la cantidad de espacio utilizado, incluido todo el espacio consumido por descendientes, como los sistemas de archivos y las instantáneas. La configuración de una cuota en un descendiente de un conjunto de datos que ya tiene una no anula la cuota del descendiente, sino que impone un límite adicional. Las cuotas no se pueden establecer en volúmenes, ya que la propiedad `volsize` representa una cuota implícita. Para obtener información sobre la configuración de cuotas, consulte "Establecimiento de cuotas en sistemas de archivos ZFS" en la página 218.
`readonly`	Booleano	`off`	Controla si un conjunto de datos se puede modificar. Si se establece en `on`, no se pueden efectuar modificaciones. La abreviatura de la propiedad es `rdonly`.
`recordsize`	Número	`128K`	Especifica un tamaño de bloque sugerido para los archivos del sistema de archivos. La abreviatura de la propiedad es `recsize`. Para obtener información detallada, consulte "Propiedad `recordsize`" en la página 201.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`referenced`	Número	N/D	Propiedad de sólo lectura que identifica la cantidad de datos a los que puede acceder un conjunto de datos, y que se pueden compartir o no con otros conjuntos de datos de la agrupación. Cuando se crea una instantánea o un clon, inicialmente hace referencia a la misma cantidad de espacio en el disco que la instantánea o el sistema de archivos del que se creó, porque su contenido es idéntico. La abreviatura de la propiedad es `refer`.
`refquota`	Número (o none)	`none`	Establece la cantidad de espacio en el disco que puede consumir un conjunto de datos. Esta propiedad impone un límite físico en la cantidad de espacio que se usa. El límite físico no incluye el espacio que utilizan los descendientes, como las instantáneas y los clones.
`refreservation`	Número (o none)	`none`	Establece la cantidad mínima de espacio en el disco que se garantiza a un conjunto de datos, sin incluir descendientes como las instantáneas o los clones. Cuando la cantidad de espacio en el disco utilizado aparece bajo este valor, se considera que el conjunto de datos utiliza la cantidad de espacio especificado por `refreservation`. La reserva de `refreservation` se representa mediante el espacio en el disco utilizado del conjunto de datos principal, y repercute en las reservas y cuotas del conjunto de datos principal. Si se establece `refreservation`, sólo se permite una instantánea en caso de que, fuera de esta reserva, exista espacio libre en la agrupación para alojar la cantidad actual de bytes a los que se hace *referencia* en el conjunto de datos. La abreviatura de la propiedad es `refreserv`.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`reservation`	Número (o none)	`none`	Establece la cantidad de espacio mínimo en el disco garantizada para un conjunto de datos y sus descendientes. Cuando la cantidad de espacio utilizado aparece bajo este valor, se considera que el conjunto de datos utiliza la cantidad de espacio especificado por su reserva. Las reservas representan el espacio en el disco utilizado del conjunto de datos principales, y repercuten en las reservas y cuotas del conjunto de datos principales. La abreviatura de la propiedad es `reserv`. Para obtener más información, consulte "Establecimiento de reservas en sistemas de archivos ZFS" en la página 221.
`secondarycache`	Cadena	`all`	Controla la información que se almacena en la memoria caché secundaria (L2ARC). Los valores posibles son `all`, `none` y `metadata`. Si se establece en `all`, los datos de usuario y los metadatos se almacenan en la memoria caché. Si se establece en `none`, ni los datos de usuario ni los metadatos se almacenan en la memoria caché. Si se establece en `metadata`, sólo los metadatos se almacenan en la memoria caché.
`setuid`	Booleano	`on`	Controla si el bit de `setuid` se cumple en un sistema de archivos.
`shareiscsi`	Cadena	off	Controla si un volumen de ZFS se comparten como un destino iSCSI. Los valores de la propiedad son `on`, `off` y `type=disk`. Es posible que desee establecer `shareiscsi=en` para un sistema de archivos para que todos los volúmenes de ZFS dentro del sistema de archivos se comparten de forma predeterminada. Sin embargo, si establece este propiedad de un sistema de archivos, no logrará un efecto directo.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`sharenfs`	Cadena	`off`	Controla si el sistema de archivos está disponible en NFS y las opciones que se utilizan. Si se establece en `on`, el comando `zfs share` se invoca sin opciones. De lo contrario, el comando `zfs share` se invoca con opciones equivalentes al contenido de esta propiedad. Si se establece en `off`, el sistema de archivos se administra mediante los comandos heredados `share` y `unshare`, y el archivo `dfstab`. Para obtener más información sobre cómo compartir los sistemas de archivos ZFS, consulte "Cómo compartir y anular la compartición de sistemas de archivos ZFS" en la página 215.
`snapdir`	Cadena	`hidden`	Controla si el directorio `.zfs` está oculto o visible en la raíz del sistema de archivos. Para obtener más información sobre el uso de instantáneas, consulte "Información general de instantáneas de ZFS" en la página 225.
`type`	Cadena	N/D	Propiedad de sólo lectura que identifica el tipo de datos como `filesystem` (sistema de archivos o clon), `volume` o `snapshot`.
`used`	Número	N/D	Propiedad de sólo lectura que identifica la cantidad de espacio que consumen el conjunto de datos y todos sus descendientes. Para obtener información detallada, consulte "Propiedad `used`" en la página 199.
`usedbychildren`	Número	`off`	Propiedad de sólo lectura que identifica la cantidad de espacio en el disco utilizado por subordinados de este conjunto de datos, que se liberaría si todos los subordinados del conjunto de datos se destruyeran. La abreviatura de la propiedad es `usedchild`.
`usedbydataset`	Número	`off`	Propiedad de sólo lectura que identifica la cantidad de espacio en el disco que utiliza este conjunto de datos en sí, que se liberaría si se destruyera el conjunto de datos, después de eliminar primero las instantáneas y los `refreservation`. La abreviatura de la propiedad es `usedds`.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`usedbyrefreservation`	Número	`off`	Propiedad de sólo lectura que identifica la cantidad de espacio en el disco que utiliza un `refreservation` establecido en un conjunto de datos, que se liberaría si el `refreservation` se eliminara. La abreviatura de la propiedad es `usedrefreserv`.
`usedbysnapshots`	Número	`off`	Propiedad de sólo lectura que identifica la cantidad de espacio en el disco que consumen las instantáneas de este conjunto de datos. En concreto, es la cantidad de espacio en el disco que se liberaría si todas las instantáneas de este conjunto de instantáneas se destruyeran. Tenga en cuenta que esto no es simplemente la suma de las propiedades `used` de las instantáneas, ya que varias instantáneas pueden compartir el espacio. La abreviatura de la propiedad es `usedsnap`.
`version`	Número	N/D	Identifica la versión de disco de un sistema de archivos, que es independiente de la versión de la agrupación. Esta propiedad sólo se puede establecer en una versión posterior que está disponible en la versión del software admitida. Para obtener más información, consulte el comando `zfs upgrade`.
`volsize`	Número	N/D	En el caso de volúmenes, especifica el tamaño lógico del volumen. Para obtener información detallada, consulte "Propiedad `volsize`" en la página 202.
`volblocksize`	Número	`8 KB`	En volúmenes, especifica el tamaño del bloque del volumen. El tamaño del bloque no se puede cambiar cuando el volumen se ha escrito, por lo que debe establecer el tamaño del bloque en el momento de la creación del volumen. El tamaño de bloque predeterminado para volúmenes es de 8 Kbytes. Es válida cualquier potencia de 2 desde 512 bytes hasta 128 Kbytes. La abreviatura de la propiedad es `volblock`.

TABLA 6–1 Descripciones de propiedades nativas de ZFS *(Continuación)*

Nombre de propiedad	Tipo	Valor predeterminado	Descripción
`zoned`	Booleano	N/D	Indica si este conjunto de datos se ha agregado a una zona no global. Si se establece esta propiedad, el punto de montaje no recibe permisos en la zona global y ZFS no puede montar dicho sistema de archivos cuando se solicite. Al instalarse una zona por primera vez, esta propiedad se establece para cualquier sistema de archivos que se agregue. Para obtener más información sobre el uso de ZFS con zonas instaladas, consulte "Uso de ZFS en un sistema Solaris con zonas instaladas" en la página 284.
`xattr`	Booleano	`on`	Indica si los atributos extendidos se habilitan (`on`) o deshabilitan (`off`) para este sistema de archivos.

Propiedades nativas de sólo lectura de ZFS

Las propiedades nativas de sólo lectura se pueden recuperar, pero no definir. Las propiedades nativas de sólo lectura no se heredan. Algunas propiedades nativas son específicas de un tipo concreto de conjunto de datos. En tales casos, el tipo de conjunto de datos se menciona en la descripción de la Tabla 6–1.

Las propiedades nativas de sólo lectura aparecen aquí y se describen en la Tabla 6–1.

- `available`
- `compressratio`
- `creation`
- `mounted`
- `origin`
- `referenced`
- `type`
- `used`

  Para obtener más información, consulte "Propiedad `used`" en la página 199.
- `usedbychildren`
- `usedbydataset`
- `usedbyrefreservation`
- `usedbysnapshots`

Para obtener más información sobre el cálculo de espacio en el disco, incluidas las propiedades `used`, `referenced` y `available`, consulte "Cálculo del espacio de ZFS" en la página 62.

Propiedad used

La propiedad used es una propiedad de sólo lectura que identifica la cantidad de espacio en el disco que consume este conjunto de datos y todos sus descendientes. Este valor se comprueba con la cuota del conjunto de datos y la reserva. El espacio utilizado no incluye la reserva del conjunto de datos, pero considera la reserva de cualquier conjunto de datos descendiente. La cantidad de espacio que un conjunto de datos consume en su elemento principal, y la cantidad de espacio en el disco que se libera si el conjunto de datos se destruye repetidamente, es la mayor entre su espacio utilizado y su reserva.

Cuando se crean instantáneas, su espacio en el disco se comparte inicialmente entre la instantánea y el sistema de archivos, y posiblemente con instantáneas anteriores. Conforme cambia el sistema de archivos, el espacio en el disco que se compartía anteriormente se vuelve exclusivo para la instantánea, y se cuenta en el espacio utilizado de la instantánea. El espacio que utiliza una instantánea representa sólo sus datos exclusivos. Asimismo, eliminar instantáneas puede aumentar la cantidad de espacio en el disco exclusivo para (y utilizado por) otras instantáneas. Para obtener información sobre los problemas de espacio y las instantáneas, consulte "Comportamiento de falta de espacio" en la página 62.

La cantidad de espacio en el disco utilizado, disponible y con referencia no incluye los cambios pendientes. Los cambios pendientes suelen calcularse en pocos segundos. Si se confirma un cambio en un disco mediante la función `fsync(3c)` u `O_SYNC`, no se garantiza necesariamente que la información de uso del espacio en el disco se actualice de inmediato.

La información de las propiedades `usedbychildren`, `usedbydataset` , `usedbyrefreservation` y `usedbysnapshots` se puede mostrar mediante el comando `zfs list - o space`. Estas propiedades identifican la propiedad `used` en espacio en el disco que consumen los descendientes. Para obtener más información, consulte la Tabla 6–1.

Propiedades nativas de ZFS configurables

Las propiedades nativas configurables son aquellas cuyos valores se pueden recuperar y establecer. Las propiedades nativas configurables se establecen mediante el comando `zfs set`, como se describe en "Configuración de propiedades de ZFS" en la página 206, o mediante el comando `zfs create`, como se describe en "Creación de un sistema de archivos ZFS" en la página 186. Salvo las cuotas y las reservas, las propiedades nativas configurables se heredan. Si desea más información sobre cuotas y reservas, consulte "Configuración de cuotas y reservas de ZFS" en la página 217.

Algunas propiedades nativas configurables son específicas de un tipo concreto de conjunto de datos. En estos casos, el tipo de conjunto de datos concreto se menciona en la descripción de la Tabla 6–1. Si no se menciona específicamente, una propiedad se aplica a todos los tipos de conjuntos de datos: sistemas de archivos, clones, volúmenes e instantáneas.

Las propiedades configurables aparecen aquí y se describen en la Tabla 6–1.

- `aclinherit`

  Para obtener una descripción detallada, consulte "Propiedades de ACL" en la página 246.
- `aclmode`

  Para obtener una descripción detallada, consulte "Propiedades de ACL" en la página 246.
- `atime`
- `canmount`
- `checksum`
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`

  Para obtener información detallada, consulte "Propiedad `recordsize`" en la página 201.
- `refquota`
- `refreservation`
- `reservation`
- `secondarycache`
- `shareiscsi`
- `sharenfs`
- `setuid`
- `snapdir`
- `version`
- `volsize`

  Para obtener información detallada, consulte "Propiedad `volsize`" en la página 202.
- `volblocksize`
- `zoned`
- `xattr`

Propiedad canmount

Si esta propiedad se establece en off, el sistema de archivos no se puede montar mediante los comandos zfs mount ni zfs mount -a. Establecer esta propiedad en of es como establecer la propiedad mountpoint en none, excepto que el conjunto de datos todavía tiene una propiedad mountpoint normal que se puede heredar. Por ejemplo, puede establecer esta propiedad en off, así como establecer propiedades heredables para los sistemas de archivos descendientes. Sin embargo, el sistema de archivos principal no se puede montar nunca, ni los usuarios pueden acceder a él. En este caso, el sistema de archivos principal sirve como *contenedor* para poder establecer propiedades en el contenedor, pero nunca se puede acceder al contenedor en sí.

En el ejemplo siguiente, se crea userpool y su propiedad canmount se establece en off. Los puntos de montaje para los sistemas de archivos de usuario descendientes se establecen en un punto de montaje común, /export/home. Los sistemas de archivo descendientes heredan las propiedades que se establecen en el sistema de archivos superior, pero el sistema de archivos superior no se monta nunca.

```
# zpool create userpool mirror c0t5d0 c1t6d0
# zfs set canmount=off userpool
# zfs set mountpoint=/export/home userpool
# zfs set compression=on userpool
# zfs create userpool/user1
# zfs create userpool/user2
# zfs mount
userpool/user1                  /export/home/user1
userpool/user2                  /export/home/user2
```

Si la propiedad canmount se establece en noauto, el conjunto de datos sólo se puede montar de manera explícita, no automáticamente. Este valor lo utiliza el software de actualización de Oracle Solaris para que, en el momento del inicio, sólo se monten los conjuntos de datos pertenecientes al entorno de inicio activo.

Propiedad recordsize

La propiedad recordsize especifica un tamaño de bloque sugerido para los archivos del sistema de archivos.

Esta propiedad se designa exclusivamente para utilizarse con cargas de trabajo de la base de datos que acceden a los archivos en registros de tamaño fijo. ZFS ajusta automáticamente el tamaño de los bloques de acuerdo con algoritmos internos optimizados para los patrones de acceso habituales. En cuanto a las bases de datos que crean archivos muy grandes pero que acceden a los archivos en pequeños bloques aleatorios, estos algoritmos quizá funcionen por debajo de su nivel habitual. Si se especifica un valor de recordsize mayor o igual que el tamaño de grabación de la base de datos, el rendimiento puede mejorar considerablemente. El uso de esta propiedad se desaconseja de manera especial en los sistemas de archivos de finalidad general; puede afectar negativamente al rendimiento. El tamaño especificado debe ser una

potencia de 2 mayor o igual que 512 y menor o igual que 128 KB. El cambio del valor `recordsize` en los sistemas de archivos sólo afecta a los archivos creados posteriormente. No afecta a los archivos ya creados.

La abreviatura de la propiedad es `recsize`.

Propiedad `volsize`

La propiedad `volsize` especifica el tamaño lógico del volumen. De forma predeterminada, la creación de un volumen establece una reserva para la misma cantidad. Cualquier cambio en `volsize` se refleja en un cambio equivalente en la reserva. Estas comprobaciones se utilizan para evitar un comportamiento inesperado para los usuarios. Un volumen que contenga menos espacio del que indica como disponible puede provocar un comportamiento indefinido o daño de los datos, según cómo se utilice el volumen. Estos efectos también pueden darse si el tamaño del volumen se cambia durante su uso, especialmente si se reduce el tamaño. Al ajustar el tamaño del volumen se debe ir con sumo cuidado.

Aunque no se recomienda, puede crear un volumen disperso si especifica el indicador `-s` en el comando `zfs create -V` o si cambia la reserva después de crear el volumen. Un *volumen disperso* se define como un volumen donde la reserva no es igual al tamaño del volumen. En un volumen disperso, los cambios en `volsize` no se reflejan en la reserva.

Para obtener más información sobre el uso de volúmenes, consulte "Volúmenes de ZFS" en la página 281.

Propiedades de usuario de ZFS

Además de las propiedades nativas, ZFS es compatible con las propiedades aleatorias del usuario. Las propiedades del usuario no repercuten en el comportamiento del sistema de archivos ZFS, pero puede usarlas para anotar información de manera que tenga sentido en su entorno.

Los nombres de propiedad del usuario deben ajustarse a las características siguientes:

- Deben contener un signo de dos puntos (': ') para distinguirlos de las propiedades nativas.
- Además, deben contener letras minúsculas, números o los signos de puntuación siguientes: ':', '+', '.', '_'.
- La longitud máxima de un nombre de propiedad de usuario es 256 caracteres.

La convención habitual es que el nombre de la propiedad se divida en los dos componentes siguientes, pero este espacio de nombre no lo aplica ZFS:

module:property

Cuando haga un uso programático de las propiedades del usuario, utilice un nombre de dominio DNS inverso para el componente *módulo* de nombres de propiedades con vistas a reducir la posibilidad de que dos paquetes desarrollados independientemente utilicen el mismo nombre de propiedad para fines diferentes. Los nombres de propiedad que comiencen por `com.sun.` se reservan para su uso por Oracle Corporation.

Los valores de las propiedades de usuario deben ajustarse a las convenciones siguientes:

- Deben constar de cadenas aleatorias que se heredan siempre y que nunca se validan.
- La longitud máxima de la propiedad de usuario es 1024 caracteres.

Por ejemplo:

```
# zfs set dept:users=finance userpool/user1
# zfs set dept:users=general userpool/user2
# zfs set dept:users=itops userpool/user3
```

Todos los comandos que se utilizan en propiedades, como `zfs list`, `zfs get`, `zfs set`, etc., se pueden utilizar para manipular las propiedades nativas y las del usuario.

Por ejemplo:

```
zfs get -r dept:users userpool
NAME            PROPERTY    VALUE           SOURCE
userpool        dept:users  all             local
userpool/user1  dept:users  finance         local
userpool/user2  dept:users  general         local
userpool/user3  dept:users  itops           local
```

Para borrar una propiedad de usuario, utilice el comando `zfs inherit.` Por ejemplo:

```
# zfs inherit -r dept:users userpool
```

Si la propiedad no se define en ningún conjunto de datos superior, se elimina por completo.

Consulta de información del sistema de archivos ZFS

El comando `zfs list` ofrece un mecanismo ampliable para ver y consultar información del conjunto de datos. En esta sección se explican las consultas básicas y complejas.

Visualización de información básica de ZFS

Puede visualizar información básica del conjunto de datos mediante el comando `zfs list` sin opciones. Este comando muestra los nombres de todos los conjuntos de datos en el sistema y los de sus propiedades `used`, `available`, `referenced` y `mountpoint`. Para obtener más información sobre estas propiedades, consulte "Introducción a las propiedades de ZFS" en la página 189.

Por ejemplo:

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
pool                   476K  16.5G    21K  /pool
pool/clone              18K  16.5G    18K  /pool/clone
pool/home              296K  16.5G    19K  /pool/home
pool/home/marks        277K  16.5G   277K  /pool/home/marks
pool/home/marks@snap      0      -   277K  -
pool/test               18K  16.5G    18K  /test
```

También puede utilizar este comando para visualizar conjuntos de datos específicos si proporciona el nombre del conjunto de datos en la línea de comandos. Asimismo, utilice la opción `-r` para mostrar repetidamente todos los descendientes del conjunto de datos. Por ejemplo:

```
# zfs list -r pool/home/marks
NAME                   USED  AVAIL  REFER  MOUNTPOINT
pool/home/marks        277K  16.5G   277K  /pool/home/marks
pool/home/marks@snap      0      -   277K  -
```

Puede utilizar el comando `lista zfs` con el punto de montaje de un sistema de archivos. Por ejemplo:

```
# zfs list /pool/home/marks
NAME              USED  AVAIL  REFER  MOUNTPOINT
pool/home/marks   277K  16.5G   277K  /pool/home/marks
```

El ejemplo siguiente muestra cómo visualizar información básica sobre `tank/home/chua` y todos sus conjuntos de datos descendientes.

```
# zfs list -r tank/home/chua
NAME                         USED  AVAIL  REFER  MOUNTPOINT
tank/home/chua              26.0K  4.81G  10.0K  /tank/home/chua
tank/home/chua/projects       16K  4.81G   9.0K  /tank/home/chua/projects
tank/home/chua/projects/fs1    8K  4.81G     8K  /tank/home/chua/projects/fs1
tank/home/chua/projects/fs2    8K  4.81G     8K  /tank/home/chua/projects/fs2
```

Para obtener más información sobre el comando `zfs list`, consulte `zfs(1M)`.

Creación de consultas de ZFS complejas

La salida `zfs list` se puede personalizar mediante las opciones `-o`, `-t` y `-H`.

Puede personalizar la salida del valor de las propiedades mediante la opción `-o` y una lista separada por comas de las propiedades en cuestión. También puede proporcionar una propiedad del conjunto de datos como argumento válido. Para obtener una lista de todas las propiedades de conjuntos de datos compatibles, consulte "Introducción a las propiedades de ZFS" en la página 189. Además de las propiedades que se definen, la lista de la opción `-o` también puede contener el `name` literal para indicar que la salida debe incluir el nombre del conjunto de datos.

El ejemplo siguiente utiliza `zfs list` para mostrar el nombre del conjunto de datos, junto con los valores de las propiedades `sharenfs` y `mountpoint` .

```
# zfs list -o name,sharenfs,mountpoint
NAME                  SHARENFS         MOUNTPOINT
tank                  off              /tank
tank/home             on               /tank/home
tank/home/ahrens      on               /tank/home/ahrens
tank/home/bonwick     on               /tank/home/bonwick
tank/home/chua        on               /tank/home/chua
tank/home/eschrock    on               legacy
tank/home/moore       on               /tank/home/moore
tank/home/tabriz      ro               /tank/home/tabriz
```

Puede utilizar la opción `-t` para especificar los tipos de conjuntos de datos que se deben mostrar. Los tipos válidos se describen en la tabla siguiente.

TABLA 6–2 Tipos de conjuntos de datos de ZFS

Tipo	Descripción
`filesystem`	Sistemas de archivos y clones
`volume`	Volúmenes
`snapshot`	Instantáneas

Las opciones `-t` toman una lista separada por comas de los tipos de conjuntos de datos que mostrar. El ejemplo siguiente utiliza las opciones `-t` y `-o` simultáneamente para mostrar el nombre y la propiedad `used` para todos los sistemas:

```
# zfs list -t filesystem -o name,used
NAME              USED
pool              476K
pool/clone         18K
pool/home         296K
pool/home/marks   277K
pool/test          18K
```

Puede utilizar la opción `-H` para omitir la cabecera `zfs list` de la salida que se ha generado. Con la opción `-H`, todos los espacios en blanco se sustituyen por el carácter de tabulación. Puede usar esta opción si necesita una salida analizable; por ejemplo, con las secuencias de comandos. El ejemplo siguiente muestra la salida generada a partir del uso del comando `zfs list` con la opción `-H`:

```
# zfs list -H -o name
pool
pool/clone
pool/home
pool/home/marks
pool/home/marks@snap
pool/test
```

Administración de propiedades de ZFS

Las propiedades del conjunto de datos se administran mediante los subcomandos `set`, `inherit` y `get` del comando `zfs`.

Configuración de propiedades de ZFS

Puede utilizar el comando `zfs set` para modificar cualquier propiedad configurable del conjunto de datos. También puede usar el comando `zfs create` para establecer las propiedades cuando se crea el conjunto de datos. Para obtener una lista de propiedades del conjunto de datos configurables, consulte "Propiedades nativas de ZFS configurables" en la página 199.

El comando `zfs set` toma una secuencia de propiedad/valor con el formato de *propiedad=valor* y un nombre de conjunto de datos. Sólo se puede establecer o modificar una propiedad durante cada invocación de `zfs set`.

El ejemplo siguiente establece la propiedad `atime` en `off` para `tank/home`.

```
# zfs set atime=off tank/home
```

Además, cualquier propiedad del sistema de archivos se puede establecer al crear el sistema. Por ejemplo:

```
# zfs create -o atime=off tank/home
```

Puede especificar valores numéricos de propiedades mediante el uso de sufijos sencillos (en orden creciente de importancia): `BKMGTPEZ`. Cualquiera de estos sufijos puede ir seguido de una `b` opcional que indica los bytes, con la excepción del sufijo `B`, que ya indica los bytes. Las cuatro invocaciones siguientes de `zfs set` son expresiones numéricas equivalentes que indican que la propiedad `quota` se puede establecer en el valor de 50 GB en el sistema de archivos `tank/home/marks`:

```
# zfs set quota=50G tank/home/marks
# zfs set quota=50g tank/home/marks
# zfs set quota=50GB tank/home/marks
# zfs set quota=50gb tank/home/marks
```

Los valores de propiedades no numéricas distinguen mayúsculas de minúsculas y deben estar en minúsculas, excepto `mountpoint` y `sharenfs`. Los valores de estas propiedades pueden tener caracteres en mayúscula y minúscula.

Para obtener más información sobre el comando `zfs set`, consulte `zfs(1M)`.

Herencia de propiedades de ZFS

Todas las propiedades configurables, con la excepción de cuotas y reservas, heredan el valor del conjunto de datos superior, a menos que en el descendiente se establezca explícitamente una cuota o reserva. Si ningún superior tiene un valor explícito establecido para una propiedad heredada, se usa el valor predeterminado para la propiedad. Puede utilizar el comando `zfs inherit` para eliminar un valor de propiedad y, de este modo, hacer que el valor se herede del elemento superior.

El ejemplo siguiente utiliza el comando `zfs set` para activar la compresión para el sistema de archivos `tank/home/bonwick`. A continuación, `zfs inherit` se utiliza para desconfigurar la propiedad `compression`; de este modo, la propiedad hereda el valor predeterminado de `off`. Como ni `home` ni `tank` tienen la propiedad `compression` configurada localmente, se utiliza el valor predeterminado. Si ambos tienen activada la compresión, se utiliza el valor configurado en el superior más inmediato (`home` en este ejemplo).

```
# zfs set compression=on tank/home/bonwick
# zfs get -r compression tank
NAME              PROPERTY      VALUE                  SOURCE
tank              compression   off                    default
tank/home         compression   off                    default
tank/home/bonwick compression   on                     local
# zfs inherit compression tank/home/bonwick
# zfs get -r compression tank
NAME              PROPERTY      VALUE                  SOURCE
tank              compression   off                    default
tank/home         compression   off                    default
tank/home/bonwick compression   off                    default
```

El subcomando `inherit` se aplica de forma recursiva cuando se especifica la opción `-r`. En el ejemplo siguiente, el comando hace que el valor de la propiedad `compression:` sea heredado por `tank/home` y cualquier descendiente que pudiera haber:

```
# zfs inherit -r compression tank/home
```

Nota – Si se utiliza la opción `-r`, se borra la configuración actual de la propiedad en todos los conjuntos de datos descendientes.

Para obtener más información sobre el comando `zfs inherit`, consulte `zfs`(1M).

Consulta de las propiedades de ZFS

La forma más sencilla de consultar los valores de las propiedades es mediante el comando `zfs list`. Para obtener más información, consulte "Visualización de información básica de ZFS" en la página 203. Sin embargo, en el caso de consultas y secuencias de comandos complejas, use el comando `zfs get` para proporcionar información detallada en un formato personalizado.

Puede utilizar el comando `zfs get` para recuperar cualquier propiedad del conjunto de datos. El ejemplo siguiente muestra la manera de recuperar un solo valor de propiedad en un conjunto de datos:

```
# zfs get checksum tank/ws
NAME            PROPERTY       VALUE                      SOURCE
tank/ws         checksum       on                         default
```

La cuarta columna, `SOURCE`, indica el origen de este valor de propiedad. La tabla siguiente define los posibles valores de origen.

TABLA 6–3 Valores posibles de `SOURCE` (`zfs get`)

Valor de origen	Descripción
`default`	Este valor de propiedad nunca se ha configurado explícitamente para este conjunto de datos ni sus superiores. En esta propiedad se utiliza el valor predeterminado.
`inherited from` *nombre_conjunto_datos*	El valor de esta propiedad se hereda del superior, tal como especifica *nombre_conjunto_datos*.
`local`	El valor de esta propiedad se ha configurado explícitamente para este conjunto de datos mediante `zfs set`.
`temporary`	El valor de esta propiedad se ha establecido mediante la opción `zfs mount -o` y sólo es válida durante el ciclo de vida del montaje. Para obtener más información sobre las propiedades de puntos de montaje temporales, consulte "Uso de propiedades de montaje temporales" en la página 214.
- (none)	Esta propiedad es de sólo lectura. Su valor lo ha generado ZFS.

Puede utilizar la palabra clave especial `all` para recuperar todos los valores de propiedades del conjunto de datos. Los ejemplos siguientes usan la palabra clave `all`:

```
# zfs get all tank/home
NAME       PROPERTY              VALUE                  SOURCE
tank/home  type                  filesystem             -
tank/home  creation              Tue Jun 29 11:44 2010  -
tank/home  used                  21K                    -
tank/home  available             66.9G                  -
tank/home  referenced            21K                    -
tank/home  compressratio         1.00x                  -
tank/home  mounted               yes                    -
tank/home  quota                 none                   default
tank/home  reservation           none                   default
tank/home  recordsize            128K                   default
tank/home  mountpoint            /tank/home             default
tank/home  sharenfs              off                    default
tank/home  checksum              on                     default
tank/home  compression           off                    default
tank/home  atime                 on                     default
```

```
tank/home  devices              on                  default
tank/home  exec                 on                  default
tank/home  setuid               on                  default
tank/home  readonly             off                 default
tank/home  zoned                off                 default
tank/home  snapdir              hidden              default
tank/home  aclmode              groupmask           default
tank/home  aclinherit           restricted          default
tank/home  canmount             on                  default
tank/home  shareiscsi           off                 default
tank/home  xattr                on                  default
tank/home  copies               1                   default
tank/home  version              4                   -
tank/home  utf8only             off                 -
tank/home  normalization        none                -
tank/home  casesensitivity      sensitive           -
tank/home  vscan                off                 default
tank/home  nbmand               off                 default
tank/home  sharesmb             off                 default
tank/home  refquota             none                default
tank/home  refreservation       none                default
tank/home  primarycache         all                 default
tank/home  secondarycache       all                 default
tank/home  usedbysnapshots      0                   -
tank/home  usedbydataset        21K                 -
tank/home  usedbychildren       0                   -
tank/home  usedbyrefreservation 0                   -
tank/home  logbias              latency             default
```

Nota – Las propiedades `casesensitivity`, `nbmand`, `normalization`, `sharesmb`, `utf8only` y `vscan` no están totalmente operativas en la versión Oracle Solaris 10 porque el servicio Oracle Solaris SMB servicio no es compatible con la versión Oracle Solaris 10.

La opción `-s` de `zfs get` permite especificar, por tipo de origen, las propiedades que mostrar. Esta opción toma una lista separada por comas que indica los tipos de origen deseados. Sólo aparecen las propiedades con el tipo de origen especificado. Los tipos de origen válidos son `local`, `default`, `inherited`, `temporary` y `none`. El ejemplo siguiente muestra todas las propiedades que se han establecido localmente en `pool`.

```
# zfs get -s local all pool
NAME             PROPERTY      VALUE                     SOURCE
pool             compression   on                        local
```

Cualquiera de las opciones anteriores se puede combinar con la opción `-r` para mostrar de forma recursiva las propiedades especificadas en todos los subordinados del conjunto de datos indicado. En el ejemplo siguiente, todas las propiedades temporales de todos los conjuntos de datos en `tank` aparecen de forma recursiva:

```
# zfs get -r -s temporary all tank
NAME             PROPERTY       VALUE                     SOURCE
tank/home          atime          off                     temporary
tank/home/bonwick  atime          off                     temporary
tank/home/marks    atime          off                     temporary
```

Puede consultar los valores de las propiedades mediante el comando `zfs get` sin especificar un sistema de archivos de destino, lo cual significa que el comando funciona en todas las agrupaciones o los sistemas de archivos. Por ejemplo:

```
# zfs get -s local all
tank/home               atime          off                    local
tank/home/bonwick       atime          off                    local
tank/home/marks         quota          50G                    local
```

Para obtener más información sobre el comando `zfs get`, consulte `zfs(1M)`.

Consulta de propiedades de ZFS para secuencias de comandos

El comando `zfs get` admite las opciones `-H` y `-o`, diseñadas para secuencias de comandos. Puede utilizar la opción `-H` para omitir información de cabecera y sustituir un espacio en blanco con el carácter de tabulación. El espacio en blanco uniforme permite el fácil análisis de los datos. Puede utilizar la opción `-o` para personalizar la salida de los modos siguientes:

- El `nombre` literal se puede utilizar con una lista separada por comas de propiedades como se definen en la sección "Introducción a las propiedades de ZFS" en la página 189.
- Una lista separada por comas de los campos literales, `name`, `value`, `property`, y `source`, que deben salir seguidos por un espacio y un argumento, que es una lista separada por comas de las propiedades.

El ejemplo siguiente muestra la forma de recuperar un valor simple mediante las opciones `-H` y `-o` de `zfs get`:

```
# zfs get -H -o value compression tank/home
on
```

La opción `-p` informa de valores numéricos como sus valores exactos. Por ejemplo, 1 MB se especifica como 1000000. Esta opción puede usarse de la forma siguiente:

```
# zfs get -H -o value -p used tank/home
182983742
```

Puede utilizar la opción `-r` junto con una de las opciones anteriores para recuperar de forma recursiva los valores solicitados para todos los descendientes. El ejemplo siguiente utiliza las opciones `-H`, `-o` y `-r` para recuperar el nombre del conjunto de datos y el valor de la propiedad `used` para `export/home` y sus descendientes, mientras se omite la salida de cualquier encabezado:

```
# zfs get -H -o name,value -r used export/home
export/home             5.57G
export/home/marks       1.43G
export/home/maybee      2.15G
```

Montaje y compartición de sistemas de archivos ZFS

En esta sección se describe la forma de administrar en ZFS puntos de montaje y sistemas de archivos compartidos.

Administración de puntos de montaje de ZFS

De manera predeterminada, un sistema de archivos ZFS se monta automáticamente cuando se crea. Puede determinar un comportamiento de punto de montaje específico para un sistema de archivos, tal y como se describe en esta sección.

También puede establecer el punto de montaje predeterminado para el conjunto de datos de una agrupación al crearlo mediante la opción `m` del comando `-zpool create`. Para obtener más información sobre la creación de agrupaciones, consulte "Creación de una agrupación de almacenamiento de ZFS" en la página 72.

De forma predeterminada, todos los sistemas de archivos ZFS se montan con ZFS en el inicio mediante el servicio `svc://system/filesystem/local` de la Utilidad de gestión de servicios (SMF). Los sistemas de archivos se montan en */ruta*, donde *ruta* corresponde al nombre del sistema de archivos.

Puede anular el punto de montaje predeterminado si utiliza el comando `zfs set` para establecer la propiedad `mountpoint` en una ruta específica. ZFS crea automáticamente este punto de montaje, si fuera necesario, y monta de forma automática el sistema de archivos asociados al invocarse el comando `zfs mount -a`, sin solicitar la edición del archivo `/etc/vfstab`.

La propiedad `mountpoint` se hereda. Por ejemplo, si `pool/home` tiene la propiedad `mountpoint` configurada en `/export/stuff`, entonces `pool/home/user` hereda `/export/stuff/user` para su propiedad `mountpoint`.

Para evitar que se monte un sistema de archivos, establezca la propiedad `mountpoint` en `none`. Además, la propiedad `canmount` se puede utilizar para controlar si se puede montar un sistema de archivos. Para obtener información sobre la propiedad `canmount`, consulte "Propiedad `canmount`" en la página 201.

Los sistemas de archivos también se administran a través de las interfaces de montaje heredadas utilizando `zfs establecido` para definir la propiedad `mountpoint` en `legacy`. De este modo, se impide que ZFS monte y administre automáticamente un sistema de archivos. En su lugar se deben utilizar las herramientas heredadas que incluyen los comandos `mount` y `umount`, así como

el archivo `/etc/vfstab`. Para obtener más información sobre montajes heredados, consulte "Puntos de montaje heredados" en la página 212.

Puntos de montaje automáticos

- Cuando cambie la propiedad `mountpoint` de `legacy` o `none` a una ruta específica, ZFS monta automáticamente el sistema de archivos.
- Si ZFS administra el sistema de archivos pero éste se encuentra desmontado, y se cambia la propiedad `mountpoint`, el sistema de archivos permanece sin montar.

ZFS administra cualquier conjunto de datos cuya propiedad `mountpoint` no sea `legacy`. En el ejemplo siguiente se crea un conjunto de datos cuyo punto de montaje lo administra ZFS automáticamente:

```
# zfs create pool/filesystem
# zfs get mountpoint pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mountpoint    /pool/filesystem          default
# zfs get mounted pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mounted       yes                       -
```

También puede configurar explícitamente la propiedad `mountpoint` tal como se muestra en el ejemplo siguiente:

```
# zfs set mountpoint=/mnt pool/filesystem
# zfs get mountpoint pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mountpoint    /mnt                      local
# zfs get mounted pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mounted       yes                       -
```

Si se cambia la propiedad `mountpoint`, el sistema de archivos se desmonta automáticamente del punto de montaje anterior y se vuelve a montar en el nuevo punto de montaje. Se crean directorios de punto de montaje según sea preciso. Si ZFS no puede desmontar un sistema de archivos porque está activo, se informa de un error y se debe forzar un desmontaje manual.

Puntos de montaje heredados

Puede administrar los sistemas de archivos ZFS con herramientas heredadas si la propiedad `mountpoint` se configura como `legacy`. Los sistemas de archivos heredados se deben administrar mediante los comandos `mount` y `umount`, así como el archivo `/etc/vfstab`. ZFS no monta automáticamente sistemas de archivos heredados en el inicio, y los comandos `mount` y `umount` de ZFS no funcionan en conjuntos de datos de este tipo. Los ejemplos siguientes muestran cómo configurar y administrar un conjunto de datos de ZFS en el modo de herencia:

```
# zfs set mountpoint=legacy tank/home/eschrock
# mount -F zfs tank/home/eschrock /mnt
```

Para montar automáticamente un sistema de archivos heredado en el inicio, debe agregar una entrada al archivo `/etc/vfstab`. El ejemplo siguiente muestra el aspecto que podría tener la entrada en el archivo `/etc/vfstab`:

```
#device         device        mount           FS      fsck    mount   mount
#to mount       to fsck       point           type    pass    at boot options
#

tank/home/eschrock -        /mnt          zfs        -        yes        -
```

Las entradas `device to fsck` y `fsck pass` se establecen en `-` porque el comando `fsck` no es aplicable a los sistemas de archivos ZFS. Para obtener más información sobre integridad de datos de ZFS, consulte "Semántica transaccional" en la página 47.

Montaje de sistemas de archivos ZFS

ZFS monta automáticamente sistemas de archivos cuando éstos se crean o cuando el sistema se inicia. El uso del comando `zfs mount` sólo es necesario cuando se deben cambiar las opciones de montaje, o explícitamente montar o desmontar sistemas de archivos.

El comando `zfs mount` sin argumentos muestra todos los sistemas de archivos montados que son administrados por ZFS. No se muestran los puntos de montaje administrados de herencia. Por ejemplo:

```
# zfs mount
tank                            /tank
tank/home                       /tank/home
tank/home/bonwick               /tank/home/bonwick
tank/ws                         /tank/ws
```

Puede utilizar la opción `-a` para montar todos los sistemas de archivos ZFS administrados. Los sistemas de archivos administrados de herencia no están montados. Por ejemplo:

```
# zfs mount -a
```

De forma predeterminada, ZFS no permite el montaje en la parte superior de un directorio que no está vacío. Para forzar un montaje en la parte superior de un directorio que no está vacío, debe usar la opción `-O`. Por ejemplo:

```
# zfs mount tank/home/lalt
cannot mount '/export/home/lalt': directory is not empty
use legacy mountpoint to allow this behavior, or use the -O flag
# zfs mount -O tank/home/lalt
```

Los puntos de montaje heredados se deben administrar mediante las herramientas de herencia. Intentar usar herramientas de ZFS genera un error. Por ejemplo:

```
# zfs mount pool/home/billm
cannot mount 'pool/home/billm': legacy mountpoint
```

```
use mount(1M) to mount this filesystem
# mount -F zfs tank/home/billm
```

Cuando se monta un sistema de archivos, éste utiliza un conjunto de opciones de montaje basadas en los valores de propiedad asociados con el conjunto de datos. La correspondencia entre las propiedades y las opciones de montaje es la siguiente:

TABLA 6–4 Propiedades relacionadas con el montaje de ZFS y opciones de montaje

Propiedad	Opción de montaje
atime	Atime/noatime
devices	devices/nodevices
exec	exec/noexec
nbmand	Nbmand/nonbmand
readonly	ro/rw
setuid	setuid/nosetuid
xattr	Xattr/noaxttr

La opción de montaje `nosuid` es un alias de `nodevices,nosetuid`.

Uso de propiedades de montaje temporales

Si alguna de las opciones anteriores se configura explícitamente mediante la opción `-o` con el comando `zfs mount`, el valor de propiedad asociado se anula de manera temporal. Estos valores de propiedades se indican como `temporary` mediante el comando `zfs get` y recuperan la configuración original cuando se desmonta el sistema de archivos. Si se cambia un valor de propiedad mientras se monta el conjunto de datos, el cambio surte efecto inmediatamente y se anula cualquier configuración temporal.

En el ejemplo siguiente, la opción de montaje de sólo lectura se configura temporalmente en el sistema de archivos `tank/home/perrin`. Se supone que el sistema de archivos está desmontado.

```
# zfs mount -o ro tank/home/perrin
```

Para cambiar temporalmente una propiedad de un sistema de archivos que está montado, debe usar la opción especial `remount`. En el ejemplo siguiente, la propiedad `atime` se cambia temporalmente a `off` para un sistema de archivos que esté montado:

```
# zfs mount -o remount,noatime tank/home/perrin
# zfs get atime tank/home/perrin
NAME             PROPERTY      VALUE                      SOURCE
tank/home/perrin atime         off                        temporary
```

Para obtener más información sobre el comando `zfs mount`, consulte `zfs(1M)`.

Desmontaje de los sistemas de archivos ZFS

Los sistemas de archivos ZFS se pueden desmontar mediante el subcomando `zfs unmount`. El comando `unmount` puede considerar como argumentos el punto de montaje o el nombre del sistema de archivos.

En el ejemplo siguiente, el nombre del sistema de archivos desmonta un sistema de archivos:

```
# zfs unmount tank/home/tabriz
```

En el ejemplo siguiente, el punto de montaje desmonta el sistema de archivos:

```
# zfs unmount /export/home/tabriz
```

El comando `unmount` falla si el sistema de archivos está ocupado. Para forzar el desmontaje de un sistema de archivos, puede usar la opción `-f`. Tenga cuidado al forzar el desmontaje de un sistema de archivos si su contenido está en uso. La aplicación se puede comportar de manera imprevista.

```
# zfs unmount tank/home/eschrock
cannot unmount '/export/home/eschrock': Device busy
# zfs unmount -f tank/home/eschrock
```

Para ofrecer compatibilidad con versiones anteriores, el comando `umount` se puede usar para desmontar sistemas de archivos ZFS. Por ejemplo:

```
# umount /export/home/bob
```

Para obtener más información sobre el comando `zfs umount`, consulte `zfs(1M)`.

Cómo compartir y anular la compartición de sistemas de archivos ZFS

ZFS puede compartir automáticamente sistemas de archivos mediante la configuración de la propiedad `sharenfs`. Gracias a este método, no hay necesidad de modificar el archivo `/etc/dfs/dfstab` cuando se comparte un nuevo sistema de archivos. La propiedad `sharenfs` es una lista de opciones separada por comas para pasar al comando `share`. El valor `on` es un alias para las opciones de compartición predeterminadas, que ofrecen permisos `read/write` a cualquier usuario. El valor `off` indica que el sistema de archivos no está administrado por ZFS y se puede compartir por medios tradicionales, como el archivo `/etc/dfs/dfstab`. Todos los sistemas de archivos cuya propiedad `sharenfs` no esté establecida en `off` se comparten durante el inicio.

Control de la semántica de compartición

De forma predeterminada, todos los sistemas de archivos están sin compartir. Para compartir un nuevo sistema de archivos, utilice una sintaxis de `zfs set` similar a la siguiente:

```
# zfs set sharenfs=on tank/home/eschrock
```

La propiedad `sharenfs` se hereda y los sistemas de archivos se comparten automáticamente al crearse, si su propiedad heredada no es `off`. Por ejemplo:

```
# zfs set sharenfs=on tank/home
# zfs create tank/home/bricker
# zfs create tank/home/tabriz
# zfs set sharenfs=ro tank/home/tabriz
```

`tank/home/bricker` y `tank/home/tabriz` son inicialmente de escritura compartida porque heredan la propiedad `sharenfs` de `tank/home`. Si la propiedad se establece en `ro` (sólo lectura), `tank/home/tabriz` es de sólo lectura compartida, al margen de la propiedad `sharenfs` que se ha configurado para `tank/home` .

Anulación de sistemas de archivos ZFS compartidos

Si bien la compartición de la mayoría de los sistemas de archivos se activa o desactiva al iniciase, crearse y destruirse, en ocasiones la compartición de los sistemas de archivos se debe anular de forma explícita. Para ello, utilice el comando `zfs unshare`. Por ejemplo:

```
# zfs unshare tank/home/tabriz
```

Este comando anula la compartición del sistema de archivos `tank/home/tabriz`. Para que los sistemas de archivos ZFS dejen de compartirse en el sistema, debe usar la opción `-a`.

```
# zfs unshare -a
```

Cómo compartir sistemas de archivos ZFS

La mayor parte del tiempo, el comportamiento automático de ZFS con respecto a compartir sistemas de archivos durante el inicio y la creación es suficiente para las operaciones normales. Si por algún motivo anula la compartición de un sistema de archivos, puede compartirlo de nuevo mediante el comando `zfs share`. Por ejemplo:

```
# zfs share tank/home/tabriz
```

También puede compartir todos los sistemas de archivos ZFS en el sistema mediante la opción `-a`.

```
# zfs share -a
```

Comportamiento de compartición heredado

Si la propiedad `sharenfs` se establece en `off`, ZFS no intenta compartir ni anular la compartición del sistema de archivos en ningún momento. Este valor permite administrar la compartición de sistemas de archivos mediante medios tradicionales, como el archivo `/etc/dfs/dfstab`.

A diferencia del comando `mount` heredado, los comandos `share` y `unshare` heredados todavía son válidos en sistemas de archivos ZFS. De este modo, puede compartir manualmente un sistema de archivos con opciones distintas de las de la propiedad `sharenfs`. Se desaconseja este modelo de administración. Administre las comparticiones de NFS íntegramente con ZFS o con el archivo `/etc/dfs/dfstab`. El modelo de administración de ZFS se ha ideado para ser más sencillo y requerir menos recursos que el modelo tradicional.

Configuración de cuotas y reservas de ZFS

Puede usar la propiedad `quota` para establecer un límite en la cantidad de espacio en el disco que puede usar un sistema de archivos. Asimismo, puede usar la propiedad `reservation` para garantizar que un sistema de archivos disponga de una cierta cantidad de espacio en el disco. Ambas propiedades se aplican al conjunto de datos donde se han configurado y a todos los descendientes de ese conjunto de datos.

Es decir, si una cuota se configura en el conjunto de datos `tank/home`, la cantidad total de espacio utilizado por `tank/home` *y todos sus descendientes* no puede superar la cuota. Asimismo, si se concede una reserva a `tank/home`, `tank/home` *y todos sus descendientes* se separan de esa reserva. La propiedad `used` informa de la cantidad de espacio utilizado por un conjunto de datos y todos sus descendientes.

Las propiedades `refquota` y `refreservation` están disponibles para administrar el espacio de sistemas de archivos sin tener en cuenta el espacio en el disco que consumen los descendientes, como las instantáneas y los clones.

En esta versión de Solaris, puede establecer una cuota de *usuario* o *grupo* sobre la cantidad de espacio en el disco consumida por archivos que sean propiedad de un determinado grupo o usuario. Las propiedades de cuota de usuarios y grupos no se pueden establecer en un volumen, en un sistema de archivos que sea anterior a la versión 4, o en una agrupación que sea anterior a la versión 15.

A la hora de determinar las funciones de cuota y reserva que mejor administran los sistemas de archivos se deben tener en cuenta los puntos siguientes:

- Las propiedades `quota` y `reservation` son apropiadas para administrar el espacio en el disco consumido por conjuntos de datos y sus descendientes.
- Las propiedades `refquota` y `refreservation` son apropiadas para administrar espacio en el disco consumido por conjuntos de datos e instantáneas.

- Establecer `refquota` o `refreservation` con un valor más alto que el de quota o reservation no tiene repercusión alguna. Si establece las propiedades de `quota` o `refquota`, fallarán las operaciones que intenten exceder cualquier valor. Es posible exceder un valor de `quota` superior al de `refquota`. Si se ensucian algunos bloques de instantáneas, quizá se exceda realmente el valor de `quota` antes de exceder el valor de `refquota`.
- Las cuotas de usuarios y grupos proporcionan un medio de administrar más fácilmente el espacio en el disco con múltiples cuentas de usuario, como por ejemplo en un entorno universitario.

Para obtener más información sobre la configuración de cuotas y reservas, consulte "Establecimiento de cuotas en sistemas de archivos ZFS" en la página 218 y "Establecimiento de reservas en sistemas de archivos ZFS" en la página 221.

Establecimiento de cuotas en sistemas de archivos ZFS

Las cuotas en los sistemas de archivos ZFS se pueden configurar y visualizar mediante los comandos `zfs set` y `zfs get`. En el ejemplo siguiente, una cuota de 10 GB se establece en `tank/home/bonwick`:

```
# zfs set quota=10G tank/home/bonwick
# zfs get quota tank/home/bonwick
NAME              PROPERTY      VALUE                     SOURCE
tank/home/bonwick quota         10.0G                     local
```

Las cuotas también influyen en la salida de los comandos `zfs list` y `df`. Por ejemplo:

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
tank/home             16.5K  33.5G  8.50K  /export/home
tank/home/bonwick     15.0K  10.0G  8.50K  /export/home/bonwick
tank/home/bonwick/ws  6.50K  10.0G  8.50K  /export/home/bonwick/ws
# df -h /export/home/bonwick
Filesystem             size   used  avail capacity  Mounted on
tank/home/bonwick       10G     8K    10G     1%    /export/home/bonwick
```

Aunque `tank/home` tenga un espacio disponible en disco de 33,5 GB, `tank/home/bonwick` y `tank/home/bonwick/ws` sólo disponen cada uno de 10 GB disponibles en disco debido a la cuota en `tank/home/bonwick`.

No puede configurar una cuota con una cantidad inferior a la que esté usando un conjunto de datos. Por ejemplo:

```
# zfs set quota=10K tank/home/bonwick
cannot set quota for 'tank/home/bonwick': size is less than current used or
reserved space
```

Puede establecer un valor de `refquota` en un conjunto de datos que limite la cantidad de espacio en el disco que puede consumir el conjunto de datos. Este límite fijo no incluye el espacio en el disco consumido por descendientes. Por ejemplo:

```
# zfs set refquota=10g students/studentA
# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
profs                106K  33.2G    18K  /profs
students            57.7M  33.2G    19K  /students
students/studentA   57.5M  9.94G  57.5M  /students/studentA
# zfs snapshot students/studentA@today
# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
profs                      106K  33.2G    18K  /profs
students                  57.7M  33.2G    19K  /students
students/studentA         57.5M  9.94G  57.5M  /students/studentA
students/studentA@today       0      -  57.5M  -
```

Para mayor comodidad, puede establecer otra cuota en un conjunto de datos para administrar mejor el espacio que consumen las instantáneas. Por ejemplo:

```
# zfs set quota=20g students/studentA
# zfs list
NAME                       USED  AVAIL  REFER  MOUNTPOINT
profs                      106K  33.2G    18K  /profs
students                  57.7M  33.2G    19K  /students
students/studentA         57.5M  9.94G  57.5M  /students/studentA
students/studentA@today       0      -  57.5M  -
```

En esta situación hipotética, `studentA` puede entrar en conflicto con el límite físico de `refquota` (10 GB), pero `studentA` puede eliminar archivos que recuperar aunque haya instantáneas.

En el ejemplo anterior, la menor de las dos cuotas (10 GB si se compara con 20 GB) aparece en la salida `zfs list`. Para ver el valor de las dos cuotas, use el comando `zfs get`. Por ejemplo:

```
# zfs get refquota,quota students/studentA
NAME               PROPERTY  VALUE              SOURCE
students/studentA  refquota  10G                local
students/studentA  quota     20G                local
```

Establecimiento de las cuotas de usuarios y grupos en un sistema de archivos ZFS

Puede definir la cuota de un grupo o un usuario mediante el uso de los comandos `zfs userquota` y `zfs groupquota`, respectivamente. Por ejemplo:

```
# zfs create students/compsci
# zfs set userquota@student1=10G students/compsci
# zfs create students/labstaff
# zfs set groupquota@staff=20GB students/labstaff
```

Visualice la cuota del grupo o la del usuario actual como se indica a continuación:

```
# zfs get userquota@student1 students/compsci
NAME              PROPERTY            VALUE               SOURCE
students/compsci  userquota@student1  10G                 local
# zfs get groupquota@staff students/labstaff
```

```
NAME               PROPERTY          VALUE             SOURCE
students/labstaff  groupquota@staff  20G               local
```

Puede mostrar el uso general del espacio en el disco del usuario o grupo mediante la consulta de las propiedades siguientes:

```
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      227M   none
POSIX User  student1  455M    10G
# zfs groupspace students/labstaff
TYPE         NAME   USED  QUOTA
POSIX Group  root   217M   none
POSIX Group  staff  217M    20G
```

Para identificar el uso individual del espacio en el disco de un usuario o grupo, consulte las propiedades siguientes:

```
# zfs get userused@student1 students/compsci
NAME              PROPERTY           VALUE              SOURCE
students/compsci  userused@student1  455M               local
# zfs get groupused@staff students/labstaff
NAME               PROPERTY         VALUE            SOURCE
students/labstaff  groupused@staff  217M             local
```

Las propiedades de cuotas de grupos y usuarios no se muestran si utiliza el comando `zfs get all` *del conjunto de datos*, que muestra una lista de todas las propiedades del sistema de archivos.

Puede eliminar la cuota de un grupo o usuario como se indica a continuación:

```
# zfs set userquota@user1=none students/compsci
# zfs set groupquota@staff=none students/labstaff
```

Las cuotas de usuarios o grupos en sistemas de archivos ZFS proporcionan las siguientes funciones:

- La cuota de un usuario o grupo que se define en un sistema de archivos superior no la hereda automáticamente un sistema de archivos descendiente.
- Sin embargo, la cuota del grupo o usuario se aplica cuando se crea una instantánea o un clon a partir de un sistema de archivos que tiene una cuota de grupo o usuario. Del mismo modo, se incluye una cuota de usuario o grupo en el sistema de archivos cuando se crea una secuencia mediante el comando `zfs send`, incluso sin opción `-R`.
- Los usuarios sin privilegios sólo pueden acceder al uso de su propio espacio en el disco. El usuario raíz o el usuario al que se le haya concedido el privilegio `userused` o `groupused`, puede acceder a la información de cálculo de espacio de grupos o usuarios de todos.
- Las propiedades `userquota` y `groupquota` no se pueden establecer en volúmenes de ZFS, en un sistema de archivos anteriores a la versión 4, o en una agrupación anterior a la versión 15.

La aplicación de cuotas de usuario o grupo puede retrasarse en varios segundos. Este retraso significa que los usuarios pueden exceder su cuota antes de que el sistema perciba que se ha sobrepasado la cuota y que rechace escrituras adicionales con el mensaje de error `EDQUOT`.

Puede utilizar el comando `quota` heredado para revisar las cuotas del usuario en un entorno NFS; por ejemplo, donde se haya montado un sistema de archivos ZFS. Sin ninguna opción, el comando `quota` sólo muestra la salida si se ha superado la cuota del usuario. Por ejemplo:

```
# zfs set userquota@student1=10m students/compsci
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      227M   none
POSIX User  student1  455M    10M
# quota student1
Block limit reached on /students/compsci
```

Si reinicia la cuota de usuario y el límite de cuota ya no se supera, podrá utilizar el comando `quota -v` para revisar la cuota del usuario. Por ejemplo:

```
# zfs set userquota@student1=10GB students/compsci
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      227M   none
POSIX User  student1  455M    10G
# quota student1
# quota -v student1
Disk quotas for student1 (uid 201):
Filesystem     usage  quota  limit    timeleft  files  quota  limit    timeleft
/students/compsci
              466029 10485760 10485760
```

Establecimiento de reservas en sistemas de archivos ZFS

Una *reserva* de ZFS es una asignación de espacio en el disco de la agrupación cuya disponibilidad en un conjunto de datos está garantizada. Así, no puede reservar espacio en el disco para un conjunto de datos si ese espacio no está disponible en la agrupación. La cantidad total de todas las reservas pendientes sin consumir no puede superar la cantidad de espacio en el disco sin utilizar de la agrupación. Las reservas de ZFS se pueden configurar y visualizar mediante los comandos `zfs set` y `zfs get`. Por ejemplo:

```
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME             PROPERTY     VALUE   SOURCE
tank/home/moore  reservation  5G      local
```

Las reservas de pueden afectar a la salida del comando `zfs list`. Por ejemplo:

```
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
```

```
tank/home             5.00G  33.5G  8.50K  /export/home
tank/home/moore       15.0K  33.5G  8.50K  /export/home/moore
```

`tank/home` utiliza 5 GB de espacio, aunque la cantidad total de espacio a la que hacen referencia `tank/home` y sus descendientes es mucho menor que 5 GB. El espacio utilizado refleja el espacio reservado para `tank/home/moore`. Las reservas se tienen en cuenta en el espacio en el disco utilizado del conjunto de datos superior y se contabilizan en relación con su cuota, reserva o ambas.

```
# zfs set quota=5G pool/filesystem
# zfs set reservation=10G pool/filesystem/user1
cannot set reservation for 'pool/filesystem/user1': size is greater than
available space
```

Un conjunto de datos puede usar más espacio en el disco que su reserva, siempre que haya espacio disponible en la agrupación que no esté reservado y que el uso actual del conjunto de datos esté por debajo de su cuota. Un conjunto de datos no puede consumir espacio en el disco reservado a otro conjunto de datos.

Las reservas no son acumulativas. Es decir, una segunda invocación de `zfs set` para configurar una reserva no agrega su reserva a la que ya existe, sino que la segunda reserva sustituye la primera. Por ejemplo:

```
# zfs set reservation=10G tank/home/moore
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME             PROPERTY      VALUE                      SOURCE
tank/home/moore  reservation   5.00G                      local
```

Puede establecer una reserva `refreservation` para garantizar espacio en el disco para un conjunto de datos que no incluya espacio en el disco consumido por instantáneas y clones. Esta reserva se explica en el cálculo del espacio utilizado en los conjuntos de datos principales, y repercute en las cuotas y reservas del conjunto de datos superior. Por ejemplo:

```
# zfs set refreservation=10g profs/prof1
# zfs list
NAME                     USED  AVAIL  REFER  MOUNTPOINT
profs                   10.0G  23.2G    19K  /profs
profs/prof1               10G  33.2G    18K  /profs/prof1
```

También se puede establecer una reserva en el mismo conjunto de datos para garantizar espacio de conjunto de datos e instantáneas. Por ejemplo:

```
# zfs set reservation=20g profs/prof1
# zfs list
NAME                     USED  AVAIL  REFER  MOUNTPOINT
profs                   20.0G  13.2G    19K  /profs
profs/prof1               10G  33.2G    18K  /profs/prof1
```

Las reservas regulares se explican en el cálculo del espacio utilizado en el principal.

En el ejemplo anterior, la menor de las dos cuotas (10 GB si se compara con 20 GB) aparece en la salida `zfs list`. Para ver el valor de las dos cuotas, use el comando `zfs get`. Por ejemplo:

```
# zfs get reservation,refreserv profs/prof1
NAME         PROPERTY        VALUE        SOURCE
profs/prof1  reservation     20G          local
profs/prof1  refreservation  10G          local
```

Si se establece `refreservation`, sólo se permite una instantánea en caso de que fuera de esta reserva exista suficiente espacio no reservado en la agrupación para alojar la cantidad actual de bytes *a los que se hace referencia* en el conjunto de datos.

◆ ◆ ◆ CAPÍTULO 7

Uso de clones e instantáneas de Oracle Solaris ZFS

En este capítulo se explican procedimientos para crear y administrar clones e instantáneas de Oracle Solaris ZFS. También contiene información sobre cómo guardar instantáneas.

Este capítulo se divide en las secciones siguientes:

Información general de instantáneas de ZFS

Una *instantánea* es una copia de sólo lectura de un sistema de archivos o volumen. Las instantáneas se pueden crear de forma casi inmediata y al principio consumen poco espacio en el disco de la agrupación. Sin embargo, a medida que el conjunto de datos va cambiando, la instantánea consume espacio en el disco al seguir haciendo referencia a los datos antiguos, lo que impide la liberación de espacio.

Las instantáneas de ZFS presentan las características siguientes:

- Se mantienen en sucesivos reinicios del sistema.
- El número máximo teórico de instantáneas es 2^{64}.
- Las instantáneas no utilizan un almacén de copia de seguridad independiente. Las instantáneas consumen espacio en el disco directamente de la misma agrupación de almacenamiento que el sistema de archivos o el volumen a partir del que se crearon.

- Las instantáneas recursivas se crean rápidamente como una operación atómica Las instantáneas se crean todas juntas (todas a la vez) o no se crea ninguna. La ventaja de las operaciones atómicas de instantáneas estriba en que los datos se toman siempre en un momento coherente, incluso en el caso de sistemas de archivos descendientes.

No se puede acceder directamente a las instantáneas de volúmenes, pero se pueden clonar, hacer copias de seguridad, invertir, etc. Para obtener información sobre cómo hacer copias de seguridad de una instantánea ZFS, consulte "Envío y recepción de datos ZFS" en la página 234.

- "Creación y destrucción de instantáneas de ZFS" en la página 226
- "Visualización y acceso a instantáneas de ZFS" en la página 229
- "Restablecimiento de una instantánea ZFS" en la página 231

Creación y destrucción de instantáneas de ZFS

Las instantáneas se crean con el comando `zfs snapshot`, que toma como único argumento el nombre de la instantánea que se va a crear. El nombre de las instantáneas se asigna de la forma siguiente:

filesystem@snapname
volume@snapname

El nombre de la instantánea debe cumplir los requisitos de denominación establecidos en "Requisitos de asignación de nombres de componentes de ZFS" en la página 51.

En el ejemplo siguiente, se crea una instantánea de `tank/home/ahrens` denominada `friday`.

```
# zfs snapshot tank/home/ahrens@friday
```

Puede crear instantáneas de todos los sistemas de archivos descendientes con la opción `-r`. Por ejemplo:

```
# zfs snapshot -r tank/home@now
# zfs list -t snapshot
NAME                         USED  AVAIL  REFER  MOUNTPOINT
rpool/ROOT/zfs2BE@zfs2BE    78.3M      -  4.53G  -
tank/home@now                   0      -    26K  -
tank/home/ahrens@now            0      -   259M  -
tank/home/anne@now              0      -   156M  -
tank/home/bob@now               0      -   156M  -
tank/home/cindys@now            0      -   104M  -
```

Las instantáneas no tienen propiedades modificables. Las propiedades de conjuntos de datos no se pueden aplicar a una instantánea. Por ejemplo:

```
# zfs set compression=on tank/home/ahrens@now
cannot set compression property for 'tank/home/ahrens@now': snapshot
properties cannot be modified
```

Para destruir instantáneas se utiliza el comando `zfs destroy`. Por ejemplo:

```
# zfs destroy tank/home/ahrens@now
```

Los conjuntos de datos no se pueden destruir si tienen una instantánea. Por ejemplo:

```
# zfs destroy tank/home/ahrens
cannot destroy 'tank/home/ahrens': filesystem has children
use '-r' to destroy the following datasets:
tank/home/ahrens@tuesday
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
```

Además, si se han creado clones a partir de una instantánea, deben destruirse antes de poder destruir la instantánea.

Para obtener más información sobre el subcomando `destroy`, consulte "Destrucción de un sistema de archivos ZFS" en la página 187.

Conservación de instantáneas de ZFS

Si se implementan diferentes directivas de instantáneas automáticas de manera que `zfs receive` destruye accidentalmente las instantáneas más antiguas porque ya no existen en la parte remitente, debería considerar el uso de la función de conservación de instantáneas.

La función de *conservación* impide que se destruya una instantánea. Además, esta función permite que una instantánea con clones se elimine en espera de la eliminación del último clon mediante el comando `zfs destroy -d`. Cada instantánea tiene asociado un número de referencia de usuario, que se inicializa a cero. El número aumenta una unidad cada vez que se aplica la función de conservación a una instantánea, y se reduce una unidad cuando se anula dicha función.

En la versión anterior de Solaris, sólo era posible destruir una instantánea mediante el comando `zfs destroy` si no tenía clones. En esta versión de Solaris, la instantánea también debe tener un número de referencia cero.

Se puede aplicar la función de conservación a una instantánea o a un conjunto de ellas. Por ejemplo, la siguiente sintaxis coloca una etiqueta de conservación, `keep`, en `tank/home/cindys/snap@1` .

```
# zfs hold keep tank/home/cindys@snap1
```

Puede utilizar la opción `-r` para conservar las instantáneas de todos los sistemas de archivos descendientes. Por ejemplo:

```
# zfs snapshot -r tank/home@now
# zfs hold -r keep tank/home@now
```

Esta sintaxis agrega una sola referencia, `keep`, a la instantánea o al conjunto de instantáneas. Cada instantánea tiene su propio espacio de nombre de etiqueta y las etiquetas de conservación

deben ser exclusivas dentro de ese espacio. Si se ha aplicado la función de conservación a una instantánea, fallará cualquier intento de destruirla mediante el comando `zfs destroy`. Por ejemplo:

```
# zfs destroy tank/home/cindys@snap1
cannot destroy 'tank/home/cindys@snap1': dataset is busy
```

Para destruir una instantánea conservada, use la opción `-d`. Por ejemplo:

```
# zfs destroy -d tank/home/cindys@snap1
```

Utilice el comando `zfs holds` para ver una lista de instantáneas conservadas. Por ejemplo:

```
# zfs holds tank/home@now
NAME           TAG   TIMESTAMP
tank/home@now  keep  Thu Jul 15 11:25:39 2010

# zfs holds -r tank/home@now
NAME                   TAG   TIMESTAMP
tank/home/cindys@now  keep  Thu Jul 15 11:25:39 2010
tank/home/mark@now    keep  Thu Jul 15 11:25:39 2010
tank/home@now         keep  Thu Jul 15 11:25:39 2010
```

Puede utilizar el comando `zfs release` para eliminar la conservación de una instantánea o de un conjunto de instantáneas. Por ejemplo:

```
# zfs release -r keep tank/home@now
```

Si la instantánea se libera, se podrá destruir mediante el comando `zfs destroy`. Por ejemplo:

```
# zfs destroy -r tank/home@now
```

Hay dos nuevas propiedades de información de conservación de instantánea:

- La propiedad `defer_destroy` está `activada` si la instantánea se ha marcado para su destrucción posteriormente, mediante el comando `zfs destroy -d`. De lo contrario, la propiedad está `desactivada`.
- La propiedad `userrefs` indica el número de casos de conservación de esta instantánea, también denominados recuentos de referencia de usuario.

Cambio de nombre de instantáneas de ZFS

Se puede cambiar el nombre de las instantáneas, pero debe hacerse en la agrupación y el conjunto de datos en que se crearon. Por ejemplo:

```
# zfs rename tank/home/cindys@083006 tank/home/cindys@today
```

Además, la siguiente sintaxis de acceso directo es equivalente a la sintaxis anterior:

```
# zfs rename tank/home/cindys@083006 today
```

La siguiente operación de cambio de nombre de instantánea no es posible porque los nombres del sistema de archivos y la agrupación de destino no coinciden con los del sistema de archivos y la agrupación a partir de los cuales se creó la instantánea:

```
# zfs rename tank/home/cindys@today pool/home/cindys@saturday
cannot rename to 'pool/home/cindys@today': snapshots must be part of same
dataset
```

El comando zfs rename -r permite cambiar el nombre de instantáneas de forma recursiva. Por ejemplo:

```
# zfs list
NAME                         USED  AVAIL  REFER  MOUNTPOINT
users                        270K  16.5G    22K  /users
users/home                    76K  16.5G    22K  /users/home
users/home@yesterday            0      -    22K  -
users/home/markm              18K  16.5G    18K  /users/home/markm
users/home/markm@yesterday      0      -    18K  -
users/home/marks              18K  16.5G    18K  /users/home/marks
users/home/marks@yesterday      0      -    18K  -
users/home/neil               18K  16.5G    18K  /users/home/neil
users/home/neil@yesterday       0      -    18K  -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -r users/home
NAME                        USED  AVAIL  REFER  MOUNTPOINT
users/home                   76K  16.5G    22K  /users/home
users/home@2daysago            0      -    22K  -
users/home/markm             18K  16.5G    18K  /users/home/markm
users/home/markm@2daysago      0      -    18K  -
users/home/marks             18K  16.5G    18K  /users/home/marks
users/home/marks@2daysago      0      -    18K  -
users/home/neil              18K  16.5G    18K  /users/home/neil
users/home/neil@2daysago       0      -    18K  -
```

Visualización y acceso a instantáneas de ZFS

Puede habilitar o deshabilitar la visualización de los listados de instantáneas en la salida zfs list mediante la propiedad de agrupación listsnapshots. Esta propiedad está habilitada de forma predeterminada.

Si deshabilita esta propiedad, puede utilizar el comando zfs list -t snapshot para mostrar información de las instantáneas. O bien, habilite la propiedad de agrupación listsnapshots. Por ejemplo:

```
# zpool get listsnapshots tank
NAME  PROPERTY       VALUE      SOURCE
tank  listsnapshots  on         default
# zpool set listsnapshots=off tank
# zpool get listsnapshots tank
NAME  PROPERTY       VALUE      SOURCE
tank  listsnapshots  off         local
```

Se puede acceder a instantáneas de sistemas de archivos del directorio `.zfs/snapshot` en la raíz del sistema de archivos. Por ejemplo, si `tank/home/ahrens` se monta en `/home/ahrens`, se puede acceder a los datos de la instantánea `tank/home/ahrens@thursday` en el directorio `/home/ahrens/.zfs/snapshot/thursday` .

```
# ls /tank/home/ahrens/.zfs/snapshot
tuesday wednesday thursday
```

Se puede obtener una lista de instantáneas de la forma que se indica a continuación:

```
# zfs list -t snapshot
NAME                           USED  AVAIL  REFER  MOUNTPOINT
pool/home/anne@monday             0      -   780K  -
pool/home/bob@monday              0      -  1.01M  -
tank/home/ahrens@tuesday       8.50K      -   780K  -
tank/home/ahrens@wednesday     8.50K      -  1.01M  -
tank/home/ahrens@thursday         0      -  1.77M  -
tank/home/cindys@today         8.50K      -   524K  -
```

Se puede obtener una lista de las instantáneas creadas para un determinado sistema de archivos de la forma siguiente:

```
# zfs list -r -t snapshot -o name,creation tank/home
NAME                  CREATION
tank/home@now         Wed Jun 30 16:16 2010
tank/home/ahrens@now  Wed Jun 30 16:16 2010
tank/home/anne@now    Wed Jun 30 16:16 2010
tank/home/bob@now     Wed Jun 30 16:16 2010
tank/home/cindys@now  Wed Jun 30 16:16 2010
```

Cálculo del espacio para instantáneas de ZFS

Cuando se crea una instantánea, al principio comparte el espacio con el sistema de archivos y, posiblemente, con instantáneas antiguas. A medida que cambia el sistema de archivos, el espacio en el disco compartido inicialmente se convierte en exclusivo de la instantánea, cosa que se contabiliza como tal en la propiedad `used`. Si se eliminan instantáneas puede aumentarse la cantidad de espacio exclusivo destinado a (*usado* por) otras instantáneas.

El valor de propiedad `referenced` de espacio de una instantánea es el mismo que tenía el sistema de archivos cuando se creó la instantánea.

Puede identificar información adicional sobre el consumo de valores de la propiedad `used`. Las nuevas propiedades del sistema de archivos de sólo lectura describen el uso de espacio en el disco de clones, sistemas de archivos y volúmenes. Por ejemplo:

```
$ zfs list -o space
# zfs list -ro space tank/home
NAME                  AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
tank/home             66.3G   675M         0     26K              0       675M
tank/home@now             -      0         -       -              -          -
tank/home/ahrens      66.3G   259M         0    259M              0          0
```

```
tank/home/ahrens@now      -      0         -       -              -          -
tank/home/anne         66.3G   156M         0    156M              0          0
tank/home/anne@now        -      0         -       -              -          -
tank/home/bob          66.3G   156M         0    156M              0          0
tank/home/bob@now         -      0         -       -              -          -
tank/home/cindys       66.3G   104M         0    104M              0          0
tank/home/cindys@now      -      0         -       -              -          -
```

Para ver una descripción de estas propiedades, consulte la Tabla 6–1.

Restablecimiento de una instantánea ZFS

Puede usar el comando `zfs rollback` para anular todos los cambios efectuados en un sistema de archivos desde que se creó una instantánea concreta. El sistema de archivos vuelve al estado en que se encontraba en el momento de realizarse la instantánea. De forma predeterminada, el comando no puede restablecer una instantánea que no sea la más reciente.

Para restablecer una instantánea anterior, hay que destruir todas las instantáneas intermedias. Puede destruir versiones anteriores de instantáneas mediante la opción `-r`.

Si una instantánea intermedia tiene clones, para destruir los clones debe especificarse la opción `-R`.

Nota – El sistema de archivos que se desea restaurar se desmonta y se vuelve a montar, si actualmente está montado. Si el sistema de archivos no se puede desmontar, la restauración falla. La opción `-f` hace que se desmonte el sistema de archivos, si es necesario.

En este ejemplo, el sistema de archivos `tank/home/ahrens` se restaura a la instantánea de `tuesday`:

```
# zfs rollback tank/home/ahrens@tuesday
cannot rollback to 'tank/home/ahrens@tuesday': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
# zfs rollback -r tank/home/ahrens@tuesday
```

En este ejemplo, las instantáneas de `wednesday` y `thursday` se destruyen porque se ha restaurado la instantánea de `tuesday`.

```
# zfs list -r -t snapshot -o name,creation tank/home/ahrens
NAME                  CREATION
tank/home/ahrens@now  Wed Jun 30 16:16 2010
```

Información general sobre clones de ZFS

Un *clon* es un volumen grabable o un sistema de archivos cuyo contenido inicial es el mismo que el del conjunto de datos a partir del cual se ha creado. Al igual que sucede con las instantáneas, un clon se crea casi inmediatamente y al principio no consume espacio en el disco adicional. Asimismo, puede crear una instantánea de un clon.

Los clones sólo se pueden crear a partir de una instantánea. Al clonarse una instantánea, se crea una dependencia implícita entre ésta y el clon. Aun en el caso de que el clon se cree en alguna otra parte de la jerarquía del conjunto de datos, la instantánea original no se podrá destruir en tanto exista el clon. Al propiedad `origin` muestra esta dependencia y el comando `zfs destroy` recopila todas estas dependencias, si las hay.

Los clones no heredan las dependencias del conjunto de datos a partir del que se crean. Utilice los comandos `zfs get` y `zfs set` para ver y cambiar las propiedades de un conjunto de datos clonado. Para obtener más información sobre el establecimiento de las propiedades de conjuntos de datos de ZFS, consulte "Configuración de propiedades de ZFS" en la página 206.

Debido a que al principio un clon comparte todo su espacio en el disco con la instantánea original, el valor de su propiedad `used` se establece inicialmente en cero. A medida que se efectúan cambios en el clon, consume más espacio en el disco. La propiedad `used` de la instantánea original no incluye el espacio que consume el clon en el disco.

Creación de un clon de ZFS

Para crear un clon, utilice el comando `zfs clone`; especifique la instantánea a partir de la cual se va a crear, así como el nombre del nuevo volumen o sistema de archivos. El nuevo volumen o sistema de archivos se puede colocar en cualquier parte de la jerarquía de ZFS. El nuevo conjunto de datos es del mismo tipo (por ejemplo, volumen o sistema de archivos) que la instantánea a partir de la cual se ha creado el clon. El clon de un sistema de archivos no se puede crear en una agrupación que no sea donde se ubica la instantánea del sistema de archivos original.

En este ejemplo, se crea un clon denominado `tank/home/ahrens/bug123` con el mismo contenido inicial que la instantánea `tank/ws/gate@yesterday`:

```
# zfs snapshot tank/ws/gate@yesterday
# zfs clone tank/ws/gate@yesterday tank/home/ahrens/bug123
```

En este ejemplo, se crea un espacio de trabajo clónico a partir de la instantánea de `projects/newproject@today` para un usuario temporal denominado `projects/teamA/tempuser`. A continuación, las propiedades se establecen en el espacio de trabajo clónico.

```
# zfs snapshot projects/newproject@today
# zfs clone projects/newproject@today projects/teamA/tempuser
# zfs set sharenfs=on projects/teamA/tempuser
# zfs set quota=5G projects/teamA/tempuser
```

Destrucción de un clon de ZFS

Para destruir clones de ZFS se utiliza el comando `zfs destroy`. Por ejemplo:

```
# zfs destroy tank/home/ahrens/bug123
```

Para poder destruir la instantánea principal, antes hay que destruir los clones.

Sustitución de un sistema de archivos ZFS por un clon de ZFS

El comando `zfs promote` es apto para reemplazar un sistema de archivos ZFS activo por un clon de ese sistema de archivos. Esta función permite la clonación y sustitución de sistemas de archivos para que el sistema de archivos *original* se convierta en el clon del sistema de archivos especificado. Asimismo, posibilita la destrucción del sistema de archivos a partir del cual se creó el clon. Sin la promoción de clones no es posible destruir un sistema de archivos original de clones activos. Para obtener más información sobre la destrucción de clones, consulte "Destrucción de un clon de ZFS" en la página 233.

En este ejemplo, se clona el sistema de archivos `tank/test/productA` y el sistema de archivos clónico, `tank/test/productAbeta`, se convierte en el sistema de archivos `tank/test/productA` original.

```
# zfs create tank/test
# zfs create tank/test/productA
# zfs snapshot tank/test/productA@today
# zfs clone tank/test/productA@today tank/test/productAbeta
# zfs list -r tank/test
NAME                       USED  AVAIL  REFER  MOUNTPOINT
tank/test                  104M  66.2G    23K  /tank/test
tank/test/productA         104M  66.2G   104M  /tank/test/productA
tank/test/productA@today      0      -   104M  -
tank/test/productAbeta        0  66.2G   104M  /tank/test/productAbeta
# zfs promote tank/test/productAbeta
# zfs list -r tank/test
NAME                          USED  AVAIL  REFER  MOUNTPOINT
tank/test                     104M  66.2G    24K  /tank/test
tank/test/productA               0  66.2G   104M  /tank/test/productA
tank/test/productAbeta        104M  66.2G   104M  /tank/test/productAbeta
tank/test/productAbeta@today     0      -   104M  -
```

En esta salida `zfs list` se ha sustituido la información de cálculo de espacio en el disco del sistema de archivos `productA` original por el sistema de archivos `productAbeta`.

Puede completar el proceso de sustitución de clones cambiando el nombre de los sistemas de archivos. Por ejemplo:

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
```

Si lo desea, puede eliminar el sistema de archivos heredado. Por ejemplo:

```
# zfs destroy tank/test/productAlegacy
```

Envío y recepción de datos ZFS

El comando `zfs send` crea una representación de flujo de datos de una instantánea que se graba en una salida estándar. De forma predeterminada, se crea un flujo de datos completo. Puede redirigir la salida a un archivo u otro sistema. El comando `zfs receive` crea una instantánea cuyo contenido se especifica en el flujo de datos que figura en la entrada estándar. Si se recibe un flujo de datos completo, también se crea un sistema de archivos. Con estos comandos puede enviar y recibir datos de instantáneas ZFS y sistemas de archivos. Consulte los ejemplos de la sección siguiente.

- "Envío de una instantánea ZFS" en la página 235
- "Recepción de una instantánea ZFS" en la página 236
- "Envío y recepción de flujos de instantáneas ZFS complejos" en la página 237
- "Cómo guardar datos de ZFS con otros productos de copia de seguridad" en la página 235

Para guardar datos ZFS existen las soluciones de copia de seguridad siguientes:

- **Productos empresariales de copia de seguridad**: si necesita las siguientes funciones, considere la opción de una solución empresarial de copia de seguridad:
  - Restauración por archivo
  - Verificación de soportes de copia de seguridad
  - Administración de soportes
- **Instantáneas de sistemas de archivos y restauración de instantáneas**: utilice los comandos `zfs snapshot` y `zfs rollback` para crear con facilidad una copia de un sistema de archivos y restablecer su versión anterior si es preciso. Esta solución es válida, por ejemplo, para restaurar uno o varios archivos de una versión anterior.

  Para obtener más información sobre cómo crear y restaurar una versión de instantánea, consulte "Información general de instantáneas de ZFS" en la página 225.
- **Guardar instantáneas**: utilice los comandos `zfs send` y `zfs receive` para enviar y recibir una instantánea ZFS. Puede guardar cambios incrementales entre instantáneas, pero no puede restaurar archivos de manera individual. Es preciso restaurar toda la instantánea del sistema de archivos. Estos comandos no constituyen una solución de copia de seguridad completa para guardar los datos de ZFS.

- **Repetición remota**: utilice los comandos `zfs send` y `zfs receive` para copiar un sistema de archivos de un sistema a otro. Este proceso difiere del producto para la administración de volúmenes tradicional que quizá refleje dispositivos a través de una WAN. No se necesita ninguna clase de configuración ni hardware especiales. La ventaja de repetir un sistema de archivos ZFS es poder volver a crear un sistema de archivos de una agrupación de almacenamiento en otro sistema y especificar distintos niveles de configuración de ese nuevo conjunto, por ejemplo RAID-Z, pero con los mismos datos del sistema de archivos.
- **Utilidades de archivado**: guarde datos de ZFS con utilidades de archivado como `tar`, `cpio` y `pax`, o productos de copia de seguridad de otros proveedores. Actualmente, tanto `tar` como `cpio` traducen correctamente las listas ACL, pero no ocurre lo mismo con `pax`.

Cómo guardar datos de ZFS con otros productos de copia de seguridad

Aparte de los comandos `zfs send` y `zfs receive`, para guardar archivos ZFS también son aptas utilidades de archivado como los comandos `tar` y `cpio`. Estas utilidades permiten guardar y restaurar atributos de archivos ZFS y ACL. Seleccione las opciones correspondientes para los comandos `tar` y `cpio`.

Para obtener información actualizada sobre problemas con ZFS y productos de copia de seguridad de otros proveedores, consulte las notas de la versión de Solaris 10 o las preguntas frecuentes sobre ZFS en:

`http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq/#backupsoftware`

Envío de una instantánea ZFS

Puede utilizar el comando `zfs send` para enviar una copia de un flujo de instantáneas y recibirlo en otra agrupación del mismo sistema o en otra agrupación de un sistema diferente que se utiliza para almacenar datos de copia de seguridad. Por ejemplo, para enviar el flujo de instantáneas de otra agrupación al mismo sistema, utilice una sintaxis similar a la siguiente:

```
# zfs send tank/data@snap1 | zfs recv spool/ds01
```

Puede utilizar `zfs recv` como alias para el comando `zfs receive`.

Si envía el flujo de instantáneas a otro sistema, utilice el comando `ssh` para enviar la salida `zfs send`. Por ejemplo:

```
host1# zfs send tank/dana@snap1 | ssh host2 zfs recv newtank/dana
```

Si se envía un flujo de datos completo, no debe existir el sistema de archivos de destino.

Los datos incrementales se pueden guardar con la opción `zfs send -i`. Por ejemplo:

```
host1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh host2 zfs recv newtank/dana
```

El primer argumento (`snap1`) es la instantánea anterior y el segundo (`snap2`) la instantánea posterior. En este caso, para que la recepción incremental sea posible, debe existir el sistema de archivos `newtank/dana`.

El origen de *instantánea1* incremental se puede especificar como último componente del nombre de la instantánea. Este método abreviado significa que sólo se debe indicar el nombre después del signo de arroba `@` para *instantánea1*, que se supone que procede del mismo sistema de archivos que *instantánea2*. Por ejemplo:

```
host1# zfs send -i snap1 tank/dana@snap2 > ssh host2 zfs recv newtank/dana
```

Esta sintaxis de acceso directo es equivalente a la sintaxis incremental en el ejemplo anterior.

Si se intenta generar un flujo de datos incremental a partir de una `instantánea1` de otro sistema de archivos, aparece en pantalla el mensaje siguiente:

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Si tiene que almacenar muchas copias, puede ser conveniente comprimir una representación de flujos de datos de instantáneas de ZFS mediante el comando `gzip`. Por ejemplo:

```
# zfs send pool/fs@snap | gzip > backupfile.gz
```

Recepción de una instantánea ZFS

Al recibir una instantánea de sistema de archivos, debe tener en cuenta los aspectos siguientes:

- Se recibe tanto la instantánea como el sistema de archivos.
- Se desmontan el sistema de archivos y todos los sistemas de archivos descendientes.
- Mientras se efectúa la recepción, no es posible acceder a los sistemas de archivos.
- El sistema de archivos original que se va a recibir no debe existir mientras se transfiere.
- Si el nombre del sistema de archivos ya existe, puede utilizar el comando `zfs rename` para cambiar el nombre del sistema de archivos.

Por ejemplo:

```
# zfs send tank/gozer@0830 > /bkups/gozer.083006
# zfs receive tank/gozer2@today < /bkups/gozer.083006
# zfs rename tank/gozer tank/gozer.old
# zfs rename tank/gozer2 tank/gozer
```

Si realiza un cambio en el sistema de archivos de destino y quiere efectuar otro envío incremental de una instantánea, antes debe restaurar el sistema de archivos receptor.

Considere el siguiente ejemplo. En primer lugar, efectúe un cambio como éste en el sistema de archivos:

```
host2# rm newtank/dana/file.1
```

A continuación, realice un envío incremental de `tank/dana@snap3`. Pero antes debe restaurar la versión previa del sistema de archivos receptor para recibir la nueva instantánea incremental. O puede eliminar el paso de restauración usando la opción `-F`. Por ejemplo:

```
host1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh host2 zfs recv -F newtank/dana
```

Al recibir una instantánea incremental, ya debe existir el sistema de archivos de destino.

Si efectúa cambios en el sistema de archivos y no restaura el sistema de archivos receptor para recibir la nueva instantánea incremental, o no utiliza la opción `-F`, verá una mensaje similar a éste:

```
host1# zfs send -i tank/dana@snap4 tank/dana@snap5 | ssh host2 zfs recv newtank/dana
cannot receive: destination has been modified since most recent snapshot
```

Para que la opción `-F` funcione debidamente, primero hay que efectuar estas comprobaciones:

- Si la instantánea más reciente no coincide con el origen incremental, no se completan la restauración ni la recepción, y se genera un mensaje de error.
- Si inadvertidamente se indica un nombre de sistema de archivos que no coincide con el origen incremental especificado en el comando `zfs receive`, no se completan la restauración ni la recepción, y se genera el siguiente mensaje de error:

  ```
  cannot send 'pool/fs@name': not an earlier snapshot from the same fs
  ```

Envío y recepción de flujos de instantáneas ZFS complejos

En esta sección se describe cómo utilizar las opciones `zfs send -I` y `-R` para enviar y recibir flujos de instantáneas más complejos.

Al enviar y recibir flujos de instantáneas ZFS complejos, tenga en cuenta los puntos siguientes:

- Utilice la opción `zfs send -I` para enviar todos los flujos incrementales de una instantánea a una instantánea acumulativa. Utilice también esta opción para enviar un flujo incremental de la instantánea original para crear un clon. Para que se acepte el flujo incremental, la instantánea original ya debe estar en la parte receptora.
- Utilice la opción `zfs send -R` para enviar un flujo de repetición de todos los sistemas de archivos descendientes. Cuando se recibe el flujo de repetición, se conservan todas las propiedades, las instantáneas, los sistemas de archivos descendientes y los duplicados.
- Utilice ambas opciones para enviar un flujo de repetición incremental.
  - Se mantienen los cambios de propiedades y también las operaciones `rename` y `destroy` de instantáneas y sistemas de archivos.

- Si no se especifica `zfs recv -F` al recibir el flujo de repetición, se omiten las operaciones `destroy` de conjuntos de datos. La sintaxis `zfs recv -F` en este caso también mantiene su propiedad de aplicar *rollback (inversión) si es preciso*.
- Al igual que en otros casos (que no sean `zfs send -R`) `- i` o `-I`, si se utiliza `-I`, se envían todas las instantáneas entre `snapA` y `snapD`. Si se utiliza `-i`, sólo se envía `snapD` (para todos los descendientes).

- Para recibir cualquiera de estos nuevos tipos de flujos `zfs send`, el sistema receptor debe ejecutar una versión del software capaz de enviarlos. La versión del flujo se incrementa.

  Sin embargo, puede acceder a los flujos desde versiones de agrupaciones más antiguas utilizando una versión del software más reciente. Por ejemplo, puede enviar y recibir flujos creados con las opciones más recientes a o desde una agrupación de la versión 3. Sin embargo, debe ejecutar software reciente para recibir un flujo enviado con las opciones más recientes.

EJEMPLO 7–1 Envío y recepción de flujos de instantáneas ZFS complejos

Un grupo de instantáneas incrementales se puede combinar en una instantánea utilizando la opción `zfs send -I`. Por ejemplo:

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

Luego deberá eliminar `snapB`, `snapC` y `snapD`.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Para recibir la instantánea combinada, use el siguiente comando.

```
# zfs receive -d -F pool/fs < /snaps/fs@all-I
# zfs list
NAME                     USED  AVAIL  REFER  MOUNTPOINT
pool                     428K  16.5G    20K  /pool
pool/fs                   71K  16.5G    21K  /pool/fs
pool/fs@snapA             16K      -  18.5K  -
pool/fs@snapB             17K      -    20K  -
pool/fs@snapC             17K      -  20.5K  -
pool/fs@snapD               0      -    21K  -
```

También puede utilizar el comando `zfs send -I` para combinar una instantánea y una instantánea clónica para crear un conjunto de datos combinado. Por ejemplo:

```
# zfs create pool/fs
# zfs snapshot pool/fs@snap1
# zfs clone pool/fs@snap1 pool/clone
# zfs snapshot pool/clone@snapA
# zfs send -I pool/fs@snap1 pool/clone@snapA > /snaps/fsclonesnap-I
# zfs destroy pool/clone@snapA
# zfs destroy pool/clone
# zfs receive -F pool/clone < /snaps/fsclonesnap-I
```

EJEMPLO 7–1 Envío y recepción de flujos de instantáneas ZFS complejos *(Continuación)*

Puede utilizar el comando `zfs send -R` para repetir un sistema de archivos ZFS y todos los sistemas de archivos descendientes, hasta la instantánea en cuestión. Cuando se recibe este flujo, se conservan todas las propiedades, las instantáneas, los sistemas de archivos descendientes y los duplicados.

En el ejemplo siguiente, se crean instantáneas de los sistemas de archivos de usuario. Se crea un flujo de repetición de todas las instantáneas de usuario. A continuación, se destruyen y se recuperan las instantáneas y los sistemas de archivos originales.

```
# zfs snapshot -r users@today
# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
users                187K  33.2G    22K  /users
users@today             0      -    22K  -
users/user1           18K  33.2G    18K  /users/user1
users/user1@today       0      -    18K  -
users/user2           18K  33.2G    18K  /users/user2
users/user2@today       0      -    18K  -
users/user3           18K  33.2G    18K  /users/user3
users/user3@today       0      -    18K  -
# zfs send -R users@today > /snaps/users-R
# zfs destroy -r users
# zfs receive -F -d users < /snaps/users-R
# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
users                196K  33.2G    22K  /users
users@today             0      -    22K  -
users/user1           18K  33.2G    18K  /users/user1
users/user1@today       0      -    18K  -
users/user2           18K  33.2G    18K  /users/user2
users/user2@today       0      -    18K  -
users/user3           18K  33.2G    18K  /users/user3
users/user3@today       0      -    18K  -
```

En el ejemplo siguiente, el comando `zfs send -R` se ha usado para repetir el conjunto de datos `users` y sus descendientes y para enviar a otra agrupación el flujo repetido, `users2`.

```
# zfs create users2 mirror c0t1d0 c1t1d0
# zfs receive -F -d users2 < /snaps/users-R
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
users                 224K  33.2G    22K  /users
users@today              0      -    22K  -
users/user1            33K  33.2G    18K  /users/user1
users/user1@today      15K      -    18K  -
users/user2            18K  33.2G    18K  /users/user2
users/user2@today        0      -    18K  -
users/user3            18K  33.2G    18K  /users/user3
users/user3@today        0      -    18K  -
users2                 188K  16.5G    22K  /users2
users2@today             0      -    22K  -
users2/user1           18K  16.5G    18K  /users2/user1
```

EJEMPLO 7–1 Envío y recepción de flujos de instantáneas ZFS complejos *(Continuación)*

```
users2/user1@today      0      -    18K  -
users2/user2          18K  16.5G    18K  /users2/user2
users2/user2@today      0      -    18K  -
users2/user3          18K  16.5G    18K  /users2/user3
users2/user3@today      0      -    18K  -
```

Repetición remota de datos de ZFS

Los comandos `zfs send` y `zfs recv` se utilizan para copiar de forma remota una representación de flujos de datos de instantánea de un sistema a otro. Por ejemplo:

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Este comando envía los datos de instantánea `tank/cindy@today` y los recibe en el sistema de archivos `sandbox/restfs`. El comando también crea una instantánea `restfs@today` en el sistema `newsys`. En este ejemplo, se ha configurado al usuario para que utilice el comando `ssh` en el sistema remoto.

◆ ◆ ◆ CAPÍTULO 8

Uso de listas ACL para proteger archivos ZFS

Este capítulo proporciona información sobre cómo utilizar listas de control de acceso (ACL) para proteger los archivos ZFS. Las listas ACL proporcionan más permisos granulares que los permisos UNIX estándar.

Este capítulo se divide en las secciones siguientes:

Nuevo modelo de ACL de Solaris

Las versiones anteriores del sistema operativo Solaris admitían una implementación de ACL que se basaba principalmente en la especificación ACL de borrador POSIX. Estas clases de ACL se utilizan para proteger archivos UFS y se traducen de versiones de NFS anteriores a NFSv4.

NFSv4 es un nuevo modelo de ACL totalmente compatible que permite la interoperatividad entre clientes UNIX y que no son UNIX. La nueva implementación de ACL, tal como se indica en las especificaciones de NFSv4, aporta una semántica mucho más rica que las ACL del tipo NT.

Las diferencias principales del nuevo modelo de ACL son:

- El modelo nuevo se basa en la especificación de NFSv4 y se parece a las ACL del tipo NT.
- El nuevo modelo ofrece un conjunto mucho más granular de privilegios de acceso. Para obtener más información, consulte la Tabla 8–2.
- Las listas ACL se definen y visualizan con los comandos `chmod` e `ls`, en lugar de los comandos `setfacl` y `getfacl`.

- El modelo nuevo proporciona una mejor semántica de herencia para establecer la forma en que se aplican los privilegios de acceso del directorio a los subdirectorios, y así sucesivamente. Para obtener más información, consulte "Herencia de ACL" en la página 246.

Los modelos de ACL proporcionan un control de acceso mucho más granular que los permisos de archivos estándar. De forma muy parecida a las ACL de borrador POSIX, las nuevas ACL disponen de varias entradas de control de acceso.

Las ACL basadas en borrador POSIX emplean una sola entrada para definir los permisos que se conceden y los que se deniegan. El nuevo modelo de ACL presenta dos clases de entradas de control de acceso que afectan a la comprobación de acceso: `ALLOW` y `DENY`. Así, a partir de una entrada de control de acceso que defina un conjunto de permisos no puede deducirse si se conceden o deniegan los permisos que no se hayan definido en dicha entrada.

La traducción entre las ACL de tipo NFSv4 y las de borrador POSIX se efectúa de la manera siguiente:

- Si emplea una utilidad que tiene en cuenta ACL, por ejemplo uno de los comandos `cp`, `mv`, `tar`, `cpio` o `rcp`, para transferir archivos UFS con ACL a un sistema de archivos ZFS, las ACL de borrador POSIX se traducen a sus equivalentes del tipo NFSv4.
- Algunas ACL de tipo NFSv4 se traducen a ACL de borrador POSIX. Si una ACL de tipo NFSv4 no se traduce a una ACL de borrador POSIX, en pantalla aparece un mensaje parecido al siguiente:

  ```
  # cp -p filea /var/tmp
  cp: failed to set acl entries on /var/tmp/filea
  ```
- Si crea un contenedor UFS `tar` o `cpio` con la opción de mantener ACL (`tar -p` o `cpio -P`) en un sistema que ejecuta una versión actual de Solaris, las ACL se pierden si el contenedor se extrae a un sistema que ejecuta una versión anterior de Solaris.

  Se extraen todos los archivos con los modelos de archivos correctos, pero se omiten las entradas de ACL.
- El comando `ufsrestore` es apto para restaurar datos en un sistema de archivos ZFS. Si los datos originales incluyen las ACL de borrador POSIX, se traducen a NFSv4 ACL.
- Si intenta definir una ACL de tipo NFSv4 en un archivo UFS, en pantalla aparece un mensaje similar al siguiente:

  ```
  chmod: ERROR: ACL type's are different
  ```
- Si intenta definir una ACL de borrador POSIX en un archivo ZFS, en pantalla se muestran mensajes parecidos al siguiente:

  ```
  # getfacl filea
  File system doesn't support aclent_t style ACL's.
  See acl(5) for more information on Solaris ACL support.
  ```

Para obtener información sobre otras limitaciones con las ACL y demás productos para copias de seguridad, consulte "Cómo guardar datos de ZFS con otros productos de copia de seguridad" en la página 235.

Descripciones de la sintaxis para definir listas ACL

A continuación se detallan dos formatos básicos de ACL:

Sintaxis para definir ACL triviales

Una ACL se considera *trivial* en el sentido de que sólo representa las entradas de UNIX `owner/group/other` tradicionales.

`chmod [options] A[index]{+|=}owner@ |group@ |everyone@:`
access-permissions/.../[:*inheritance-flags*]: `deny | allow` *archivo*

`chmod [options] A-owner@, group@, everyone@:`*access-permissions*
/.../[:*inheritance-flags*]:`deny | allow` *archivo* ...

`chmod [options] A[index]-` *archivo*

Sintaxis para definir ACL no triviales

`chmod [options] A[index]{+|=}user|group:name:`*access-permissions*
/.../[:*inheritance-flags*]:`deny | allow` *archivo*

`chmod [options] A-user|group:name:`*access-permissions* /.../[:*inheritance-flags*]:`deny | allow` *archivo* ...

`chmod [options] A[index]-` *archivo*

`owner@`, `group@`, `everyone@`
: Identifica el *tipo de entrada ACL* de la sintaxis de ACL trivial. Para obtener una descripción de los tipos de entrada ACL, consulte la Tabla 8–1.

usuario o *grupo*: *ID de entrada ACL=nombre de usuario o nombre de grupo*
: Identifica el *tipo de entrada ACL* de la sintaxis de ACL explícita. El *tipo de entrada ACL* `user` y `group` también debe contener el *ID de entrada ACL*, *username* o *groupname*. Para obtener una descripción de los tipos de entrada ACL, consulte la Tabla 8–1.

access-permissions/.../
: Identifica los permisos de acceso que se conceden o deniegan. Para obtener una descripción de los permisos de acceso de ACL, consulte la Tabla 8–2.

inheritance-flags
: Identifica una lista opcional de indicadores de herencia de ACL. Para obtener una descripción de los indicadores de herencia de ACL, consulte la Tabla 8–3.

`deny` | `allow`
: Identifica si se conceden o deniegan los permisos de acceso.

En el ejemplo siguiente, el valor de *ID de entrada ACL* no es relevante:

```
group@:write_data/append_data/execute:deny
```

El ejemplo siguiente incluye un *ID de entrada ACL* porque en la ACL se incluye un usuario específico (*tipo de entrada ACL*).

```
0:user:gozer:list_directory/read_data/execute:allow
```

Cuando en pantalla se muestra una entrada de ACL, se parece al ejemplo siguiente:

```
2:group@:write_data/append_data/execute:deny
```

En este ejemplo, el **2**, también denominado *ID de índice*, identifica la entrada ACL en la ACL más grande, que podría tener varias entradas para owner (propietario), UID específicos, group (grupo) y everyone (cualquiera). Se puede especificar el *ID de índice* con el comando `chmod` para identificar la parte de la ACL que desea modificar. Por ejemplo, el ID de índice 3 puede identificarse como `A3` en la sintaxis del comando `chmod` de una forma similar a la siguiente:

```
chmod A3=user:venkman:read_acl:allow filename
```

Los tipos de entrada ACL, que son las representaciones de ACL de los propietarios, grupos, etc., se describen en la tabla siguiente.

TABLA 8–1 Tipos de entradas de ACL

Tipo de entrada de ACL	Descripción
`owner@`	Especifica el acceso que se concede al propietario del objeto.
`group@`	Especifica el acceso que se concede al grupo propietario del objeto.
`everyone@`	Especifica el acceso que se concede a cualquier usuario o grupo que no coincida con ninguna otra entrada de ACL.
`user`	Con un nombre de usuario, especifica el acceso que se concede a un usuario adicional del objeto. Esta entrada debe incluir el *ID de entrada ACL*, que contiene un *nombre_usuario* o *ID_usuario*. Si el valor no es un ID de usuario numérico o *nombre de usuario* válido, el tipo de entrada de ACL tampoco es válido.
`group`	Con un nombre de grupo, especifica el acceso que se concede a un grupo adicional del objeto. Esta entrada debe incluir el *ID de entrada ACL*, que contiene un *nombre de usuario* o *ID de grupo*. Si el valor no es un ID de grupo numérico o *nombre de grupo* válido, el tipo de entrada de ACL tampoco es válido.

En la tabla siguiente se describen los privilegios de acceso de ACL.

TABLA 8–2 Privilegios de acceso de ACL

Privilegio de acceso	Privilegio de acceso compacto	Descripción
`add_file`	`w`	Permiso para agregar un archivo nuevo a un directorio.
`add_subdirectory`	`p`	En un directorio, permiso para crear un subdirectorio.

TABLA 8–2 Privilegios de acceso de ACL *(Continuación)*

Privilegio de acceso	Privilegio de acceso compacto	Descripción
append_data	p	Marcador de posición. Actualmente no se ha implementado.
delete	d	Permiso para eliminar un archivo.
delete_child	D	Permiso para eliminar un archivo o un directorio dentro de un directorio.
execute	x	Permiso para ejecutar un archivo o buscar en el contenido de un directorio.
list_directory	r	Permiso para resumir el contenido de un directorio.
read_acl	c	Permiso para leer la ACL (`ls`).
read_attributes	a	Permiso para leer los atributos básicos (no ACL) de un archivo. Los atributos de tipo stat pueden considerarse atributos básicos. Permitir este bit de la máscara de acceso significa que la entidad puede ejecutar `ls`(1) y `stat`(2).
read_data	r	Permiso para leer el contenido de un archivo.
read_xattr	R	Permiso para leer los atributos extendidos de un archivo o al buscar en el directorio de atributos extendidos del archivo.
synchronize	s	Marcador de posición. Actualmente no se ha implementado.
write_xattr	W	Permiso para crear atributos extendidos o escribir en el directorio de atributos extendidos. Si se concede este permiso a un usuario, el usuario puede crear un directorio de atributos extendidos para un archivo. Los permisos de atributo del archivo controlan el acceso al atributo por parte del usuario.
write_data	w	Permiso para modificar o reemplazar el contenido de un archivo.
write_attributes	A	Permiso para cambiar los sellos de fecha asociados con un archivo o directorio a un valor arbitrario.
write_acl	C	Permiso para escribir en la ACL o posibilidad de modificarla mediante el comando `chmod`.
write_owner	o	Permiso para cambiar el grupo o propietario del archivo. O posibilidad de ejecutar los comandos `chown` o `chgrp` en el archivo. Permiso para adquirir la propiedad de un archivo o para cambiar la propiedad del grupo de un archivo a un grupo al que pertenezca el usuario. Si desea cambiar la propiedad de grupo o archivo a un usuario o grupo arbitrario, se necesita el privilegio `PRIV_FILE_CHOWN`.

Herencia de ACL

La finalidad de utilizar la herencia de ACL es que los archivos o directorios que se creen puedan heredar las ACL que en principio deben heredar, pero sin prescindir de los permisos en el directorio superior.

De forma predeterminada, las ACL no se propagan. Si en un directorio se establece una ACL no trivial, no se heredará en ningún directorio posterior. Debe especificar la herencia de una ACL en un archivo o directorio.

En la tabla siguiente se describen los indicadores de herencia opcionales.

TABLA 8–3 Indicadores de herencia de ACL

Indicador de herencia	Indicador de herencia compacto	Descripción
`file_inherit`	`f`	La ACL se hereda del directorio superior, pero únicamente se aplica a los archivos del directorio.
`dir_inherit`	`d`	La ACL se hereda del directorio superior, pero únicamente se aplica a los subdirectorios del directorio.
`inherit_only`	`i`	La ACL se hereda del directorio superior, pero únicamente se aplica a los archivos y subdirectorios que se creen, no al directorio en sí. Para especificar lo que se hereda. se necesita el indicador `file_inherit`, `dir_inherit` o ambos.
`no_propagate`	`n`	La ACL se hereda sólo del directorio superior al contenido del primer nivel del directorio. Se excluye el contenido del segundo nivel o inferiores. Para especificar lo que se hereda se necesita el indicador `file_inherit`, `dir_inherit` o ambos.
-	N/D	Ningún permiso concedido.

Además, se puede establecer una directriz de herencia de ACL predeterminada del sistema de archivos más o menos estricta, mediante la propiedad del sistema de archivos `aclinherit`. Para obtener más información, consulte la siguiente sección.

Propiedades de ACL

Un sistema de archivos ZFS incluye dos modos de propiedades en relación con ACL.

- `aclinherit`: propiedad que determina el comportamiento de la herencia de ACL. Puede tener los valores siguientes:
  - `discard`: en los objetos nuevos, si se crea un archivo o directorio, no se heredan entradas de ACL. La ACL del archivo o directorio nuevo es igual a los permisos del archivo o directorio.

- `noallow`: en los objetos nuevos, sólo se heredan las entradas de ACL cuyo tipo de acceso sea `deny`.
- `restricted`: en los objetos nuevos, al heredarse una entrada de ACL se eliminan los permisos `write_owner` y `write_acl`.
- `passthrough`: si el valor de propiedad se configura como `passthrough`, los archivos se crean con permisos que determinan las entradas de control de acceso que se pueden heredar. Si no existen entradas de control de acceso que se puedan heredar y que afecten a los permisos, los permisos se configurarán de acuerdo con los solicitados desde la aplicación.
- `passthrough-x` : este valor de propiedad tiene la misma semántica que `passthrough`, excepto que cuando `passthrough-x` está habilitado, los archivos se crean con el permiso de ejecución (`x`), pero sólo si este permiso se ha establecido en el modo de creación de archivos y en un ACE heredable que afecta al modo.

El valor predeterminado de la propiedad `aclinherit` es `restricted`.

- `aclmode`: esta propiedad modifica el comportamiento de ACL al crear un archivo por primera vez o si el comando `chmod` modifica los permisos de un archivo o directorio. Los valores posibles son:
  - `discard`: se eliminan todas las entradas de ACL menos las que se necesiten para definir el modo del archivo o directorio.
  - `groupmask`: se reducen los permisos de ACL de usuario o grupo para que no sean mayores que los permisos de grupo, a menos que se trate de una entrada de usuario cuyo ID de usuario sea idéntico al del propietario del archivo o directorio. Así, los permisos de ACL se reducen para que no superen los permisos del propietario.
  - `passthrough`: durante el funcionamiento de `chmod`, las entradas de control de acceso que no sean `owner@`, `group@` o `everyone@` no se modifican de ningún modo. Las entradas de control de acceso con `owner@`, `group@` o `everyone@` están desactivadas para configurar el modo de archivo de acuerdo con lo solicitado por el funcionamiento de `chmod`.

`groupmask` es el valor predeterminado de la propiedad `aclmode`.

Establecimiento de las ACL en archivos ZFS

Al implementarse con ZFS, las ACL se componen de una matriz de entradas de ACL. ZFS proporciona un modelo de ACL *puro* en el que todos los archivos disponen de una ACL. Normalmente, las ACL son *triviales* en el sentido de que únicamente representan las entradas de UNIX `owner/group/other` tradicionales.

Si cambia los permisos del archivo, su ACL se actualiza en consonancia. Además, si elimina una ACL no trivial que concedía a un usuario acceso a un archivo o directorio, ese usuario quizá siga disponiendo de acceso gracias a los bits de permisos del archivo o directorio que conceden

acceso al grupo o a todos los usuarios. Todas las decisiones de control de acceso se supeditan a los permisos representados en una ACL de archivo o directorio.

A continuación se proporcionan las reglas principales de acceso de ACL de un archivo ZFS:

- ZFS procesa entradas de ACL en el orden que figuran en la ACL, de arriba abajo.
- Sólo se procesan entradas de ACL que tengan a "alguien" que coincida con quien solicita acceso.
- Una vez se concede un permiso `allow`, una entrada `deny` de ACL posterior no lo puede denegar en el mismo conjunto de permisos de ACL.
- El permiso `write_acl` se concede de forma incondicional al propietario del archivo aunque el permiso se deniegue explícitamente. De lo contrario, se deniega cualquier permiso que no quede especificado.

  Con permisos `deny` o cuando falta un permiso de acceso para un archivo, el subsistema de privilegios determina la solicitud de acceso que se concede al propietario del archivo o superusuario. Es un mecanismo para permitir que los propietarios de archivos siempre puedan acceder a sus archivos y que los superusuarios puedan modificar archivos en situaciones de recuperación.

Si en un directorio se establece una ACL no trivial, los directorios secundarios no heredan la ACL de manera automática. Si se establece una ACL no trivial y desea que la hereden los directorios secundarios, debe utilizar los indicadores de herencia de ACL. Para obtener más información, consulte la Tabla 8–3 y "Establecimiento de herencia de ACL en archivos ZFS en formato detallado" en la página 255.

Al crear un archivo, y en función del valor `umask`, se aplica una ACL similar a la siguiente:

```
$ ls -v file.1
-rw-r--r--   1 root     root      206663 May 20 14:09 file.1
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Cada categoría de usuario (`owner@!`, `group@!`, `everyone@!`) tiene dos entradas de ACL en este ejemplo. Una entrada es para los permisos `deny` y otra para los permisos `allow`.

A continuación se proporciona una descripción de la ACL de este archivo:

`0:owner@` Se deniega al propietario el permiso de ejecución del archivo (`execute:deny`).

`1:owner@` El propietario puede leer y modificar el contenido del archivo (`read_data/write_data/append_data`). También puede modificar atributos

del archivo como indicaciones de hora, atributos extendidos y ACL (`write_xattr/write_attributes /write_acl`). Además, puede modificar la propiedad del archivo (`write_owner:allow`).

`2:group@` Se deniegan al grupo permisos de modificación y ejecución del archivo (`write_data/append_data/execute:deny`).

`3:group@` Se conceden al grupo permisos de lectura del archivo (`read_data:allow`).

`4:everyone@` Se deniegan a quien no sea propietario del archivo o miembro del grupo propietario los permisos de ejecución o modificación del contenido del archivo, y de modificación de sus atributos (`write_data/append_data/write_xattr/execute/ write_attributes/write_acl/write_owner:deny`).

`5:everyone@` Se conceden a quien no sea propietario del archivo o miembro del grupo propietario permisos de lectura del archivo y de sus atributos (`read_data/read_xattr/read_attributes/read_acl/ synchronize:allow`). El permiso de acceso `synchronize` no está implementado en la actualidad.

Al crear un directorio, y en función del valor `umask`, se aplica una ACL de directorio predeterminada similar a la siguiente:

```
$ ls -dv dir.1
drwxr-xr-x   2 root     root           2 May 20 14:11 dir.1
     0:owner@::deny
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     2:group@:add_file/write_data/add_subdirectory/append_data:deny
     3:group@:list_directory/read_data/execute:allow
     4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

A continuación se proporciona una descripción de esta ACL de directorio:

`0:owner@` La lista de `deny` del propietario está vacía para el directorio (`::deny`).

`1:owner@` El propietario puede leer y modificar el contenido del directorio (`list_directory/read_data/add_file/write_data/ add_subdirectory/append_data`), buscar en el contenido (`execute`) y modificar atributos del archivo como indicaciones de hora, atributos extendidos y ACL (`write_xattr/write_attributes/write_acl`). Además, puede modificar la propiedad del directorio (`write_owner:allow`).

`2:group@`	El grupo no puede agregar ni modificar contenido del directorio (`add_file/write_data/add_subdirectory/append_data :deny`).
`3:group@`	El grupo puede agrupar y leer el contenido del directorio. Además, tiene permisos de ejecución para buscar en el contenido del directorio (`list_directory/read_data/execute:allow`).
`4:everyone@`	Se deniega a quien no sea propietario del archivo o miembro del grupo propietario el permiso de adición o modificación del contenido del directorio (`add_file/write_data/add_subdirectory/append_data`). Además se deniega el permiso para modificar atributos del directorio (`write_xattr/write_attributes/write_acl/write_owner:deny`).
`5:everyone@`	Se conceden a quien no sea propietario del archivo o miembro del grupo propietario permisos de lectura y ejecución del contenido del directorio y sus atributos (`list_directory/read_data/read_xattr/execute/read_ attributes/read_acl/synchronize:allow`). El permiso de acceso `synchronize` no está implementado en la actualidad.

Establecimiento y visualización de ACL en archivos ZFS en formato detallado

El comando `chmod` es válido para modificar las ACL de archivos ZFS. La sintaxis siguiente del comando `chmod` para modificar ACL utiliza *especificación ACL* para identificar el formato de la ACL. Para obtener una descripción de *especificación ACL*, consulte "Descripciones de la sintaxis para definir listas ACL" en la página 243.

- Adición de entradas de ACL
  - Adición de una entrada de ACL para un usuario

    ```
    % chmod A+acl-specification  filename
    ```
  - Adición de una entrada de ACL mediante *ID_índice*

    ```
    % chmod Aindex-ID+acl-specification  filename
    ```

    Esta sintaxis inserta la nueva entrada de ACL en la ubicación de *ID_índice* que se especifica.
- Sustitución de una entrada de ACL

  ```
  % chmod A=acl-specification  filename
  ```

  ```
  % chmod Aindex-ID=acl-specification  filename
  ```
- Eliminación de entradas de ACL
  - Eliminación de una entrada de ACL mediante *ID_índice*

```
% chmod Aindex-ID- filename
```

- Eliminación de una entrada de ACL por usuario

```
% chmod A-acl-specification filename
```

- Eliminación de todas las entradas de control de acceso no triviales de un archivo

```
% chmod A- filename
```

Para ver en pantalla información de ACL en modo detallado, se utiliza el comando `ls -v`. Por ejemplo:

```
# ls -v file.1
-rw-r--r--   1 root     root      206663 May 20 14:09 file.1
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Para obtener información sobre el uso del formato de ACL compacto, consulte "Establecimiento y visualización de ACL en archivos ZFS en formato compacto" en la página 262.

EJEMPLO 8–1 Modificación de ACL triviales en archivos ZFS

Esta sección proporciona ejemplos de cómo configurar y mostrar ACL triviales.

En el ejemplo siguiente, en `file.1` hay una ACL trivial:

```
# ls -v file.1
-rw-r--r--   1 root     root      206663 May 20 15:03 file.1
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

En el ejemplo siguiente, se conceden permisos de `write_data` para `group@`:

```
# chmod A2=group@:append_data/execute:deny file.1
# chmod A3=group@:read_data/write_data:allow file.1
# ls -v file.1
-rw-rw-r--   1 root     root      206663 May 20 15:03 file.1
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
```

EJEMPLO 8–1 Modificación de ACL triviales en archivos ZFS *(Continuación)*

```
        /write_acl/write_owner:allow
     2:group@:append_data/execute:deny
     3:group@:read_data/write_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
        /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
        :allow
```

En el ejemplo siguiente, los permisos de `file.1` se establecen en `644`.

```
# chmod 644 file.1
# ls -v file.1
-rw-r--r--   1 root     root      206663 May 20 15:03 file.1
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
        /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
        /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
        :allow
```

EJEMPLO 8–2 Establecimiento de ACL no triviales en archivos ZFS

En esta sección se proporcionan ejemplos de establecimiento y visualización de ACL no triviales.

En el ejemplo siguiente, se agregan permisos de `read_data/execute` para el usuario `gozer` en el directorio `test.dir`:

```
# chmod A+user:gozer:read_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+  2 root     root           2 May 20 15:09 test.dir
     0:user:gozer:list_directory/read_data/execute:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
        /append_data/write_xattr/execute/write_attributes/write_acl
        /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
        /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
        /read_acl/synchronize:allow
```

En el ejemplo siguiente, se retiran los permisos de `read_data/execute` para el usuario `gozer`:

```
# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x   2 root     root           2 May 20 15:09 test.dir
     0:owner@::deny
```

EJEMPLO 8–2 Establecimiento de ACL no triviales en archivos ZFS *(Continuación)*

```
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     2:group@:add_file/write_data/add_subdirectory/append_data:deny
     3:group@:list_directory/read_data/execute:allow
     4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

EJEMPLO 8–3 Interacción de ACL con permisos en archivos ZFS

Estos ejemplos ilustran la interacción entre el establecimiento de ACL y el cambio de los permisos del archivo o el directorio.

En el ejemplo siguiente, en `file.2` hay una ACL trivial:

```
# ls -v file.2
-rw-r--r--   1 root     root        3103 May 20 15:23 file.2
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

En el ejemplo siguiente, se retiran los permisos `allow` de `everyone@`.

```
# chmod A5- file.2
# ls -v file.2
-rw-r-----+  1 root     root        3103 May 20 15:23 file.2
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
```

En esta salida, los permisos del archivo se restablecen de `644` a `640`. Los permisos de lectura de `everyone@` se suprimen de los permisos del archivo cuando se retiran los permisos de `allow` de ACL para `everyone@`.

En el ejemplo siguiente, la ACL se reemplaza con permisos de `read_data/write_data` para `everyone@`:

```
# chmod A=everyone@:read_data/write_data:allow file.3
# ls -v file.3
```

EJEMPLO 8–3 Interacción de ACL con permisos en archivos ZFS *(Continuación)*

```
-rw-rw-rw-+  1 root     root        6986 May 20 15:25 file.3
     0:everyone@:read_data/write_data:allow
```

En esta salida, la sintaxis de `chmod` reemplaza la ACL con permisos de `read_data/write_data:allow` por permisos de lectura/escritura para owner (propietario), group (grupo) y `everyone@` (cualquiera). En este modelo, `everyone@` especifica acceso a cualquier grupo o usuario. Como no hay entrada de ACL de `owner@` o `group@` para anular los permisos de propietario y grupo, los permisos se establecen en `666`.

En el ejemplo siguiente, la ACL se reemplaza por permisos de lectura para el usuario `gozer`:

```
# chmod A=user:gozer:read_data:allow file.3
# ls -v file.3
----------+  1 root     root        6986 May 20 15:25 file.3
     0:user:gozer:read_data:allow
```

En esta salida, los permisos de archivo se calculan como `000` porque no hay entradas de ACL para `owner@`, `group@` ni `everyone@`, que representan los componentes de permisos habituales de un archivo. El propietario del archivo puede solventar esta situación restableciendo los permisos (y la ACL) de la forma siguiente:

```
# chmod 655 file.3
# ls -v file.3
-rw-r-xr-x+  1 root     root        6986 May 20 15:25 file.3
     0:user:gozer::deny
     1:user:gozer:read_data:allow
     2:owner@:execute:deny
     3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     4:group@:write_data/append_data:deny
     5:group@:read_data/execute:allow
     6:everyone@:write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:deny
     7:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
         /synchronize:allow
```

EJEMPLO 8–4 Restauración de ACL triviales en archivos ZFS

Puede utilizar el comando `chmod` para eliminar todas las ACL no triviales de un archivo o un directorio, de modo que se restauren sus ACL triviales.

En el ejemplo siguiente, hay dos entradas de control de acceso no triviales en `test5.dir`:

```
# ls -dv test5.dir
drwxr-xr-x+  2 root     root           2 May 20 15:32 test5.dir
     0:user:lp:read_data:file_inherit:deny
     1:user:gozer:read_data:file_inherit:deny
     2:owner@::deny
     3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
```

EJEMPLO 8–4 Restauración de ACL triviales en archivos ZFS *(Continuación)*

```
        /write_owner:allow
     4:group@:add_file/write_data/add_subdirectory/append_data:deny
     5:group@:list_directory/read_data/execute:allow
     6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
        /write_attributes/write_acl/write_owner:deny
     7:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
        /read_acl/synchronize:allow
```

En el ejemplo siguiente, se han eliminado las ACL no triviales de los usuarios `gozer` y `lp`. La ACL restante contiene los seis valores predeterminados de `owner@`, `group@` y `everyone@`.

```
# chmod A- test5.dir
# ls -dv test5.dir
drwxr-xr-x   2 root     root           2 May 20 15:32 test5.dir
     0:owner@::deny
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
        /append_data/write_xattr/execute/write_attributes/write_acl
        /write_owner:allow
     2:group@:add_file/write_data/add_subdirectory/append_data:deny
     3:group@:list_directory/read_data/execute:allow
     4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
        /write_attributes/write_acl/write_owner:deny
     5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
        /read_acl/synchronize:allow
```

Establecimiento de herencia de ACL en archivos ZFS en formato detallado

Puede especificar de qué modo se heredan ACL en archivos y directorios. De forma predeterminada, las ACL no se propagan. Si en un directorio se establece una ACL no trivial, no se heredará en ningún directorio posterior. Debe establecer la herencia de una ACL en un archivo o directorio.

Además, se proporcionan dos propiedades de ACL que se pueden establecer de forma global en sistemas de archivos: `aclinherit` y `aclmode`. De modo predeterminado, `aclinherit` se establece en `restricted` y `aclmode` se establece en `groupmask`.

Para obtener más información, consulte "Herencia de ACL" en la página 246.

EJEMPLO 8–5 Concesión de herencia de ACL predeterminada

De forma predeterminada, las ACL no se propagan en una estructura de directorios.

En el ejemplo siguiente, se aplica una entrada de control de acceso no trivial de `read_data/write_data/execute` para el usuario `gozer` en el directorio `test.dir`:

EJEMPLO 8–5 Concesión de herencia de ACL predeterminada *(Continuación)*

```
# chmod A+user:gozer:read_data/write_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+  2 root     root           2 May 20 15:41 test.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Si se crea un subdirectorio de `test.dir`, no se propaga la entrada de control de acceso del usuario `gozer`. El usuario `gozer` sólo dispondrá de acceso al subdirectorio si se le conceden permisos como propietario del archivo, miembro del grupo o `everyone@`. Por ejemplo:

```
# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x   2 root     root           2 May 20 15:42 test.dir/sub.dir
     0:owner@::deny
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     2:group@:add_file/write_data/add_subdirectory/append_data:deny
     3:group@:list_directory/read_data/execute:allow
     4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

EJEMPLO 8–6 Concesión de herencia de ACL en archivos y directorios

En los ejemplos siguientes se identifican las entradas de control de acceso de archivos y directorios que se aplican al establecerse el indicador `file_inherit`.

En este ejemplo, se agregan permisos de `read_data/write_data` en archivos del directorio `test2.dir` para el usuario `gozer`, de manera que disponga de acceso de lectura en cualquier archivo que se cree:

```
# chmod A+user:gozer:read_data/write_data:file_inherit:allow test2.dir
# ls -dv test2.dir
drwxr-xr-x+  2 root     root           2 May 20 15:50 test2.dir
     0:user:gozer:read_data/write_data:file_inherit:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
```

EJEMPLO 8–6 Concesión de herencia de ACL en archivos y directorios *(Continuación)*

```
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

En el ejemplo siguiente, los permisos del usuario `gozer` se aplican en el archivo `test2.dir/file.2` recién creado. La herencia de ACL concedida, `read_data:file_inherit:allow`, significa que el usuario `gozer` puede leer el contenido de cualquier archivo que se cree.

```
# touch test2.dir/file.2
# ls -v test2.dir/file.2
-rw-r--r--+  1 root     root           0 May 20 15:51 test2.dir/file.2
     0:user:gozer:write_data:deny
     1:user:gozer:read_data/write_data:allow
     2:owner@:execute:deny
     3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     4:group@:write_data/append_data/execute:deny
     5:group@:read_data:allow
     6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Puesto que la propiedad `aclmode` para este archivo tiene el valor predeterminado, `groupmask`, el usuario `gozer` no tiene el permiso `write_data` en `file.2` porque no lo permite el permiso de grupo para el archivo.

Al usar el permiso `inherit_only`, que se concede si se establecen los indicadores `file_inherit` o `dir_inherit`, la ACL se propaga en la estructura de directorios. Así, al usuario `gozer` sólo se le conceden o deniegan permisos de `everyone@` a menos que sea propietario del archivo o miembro del grupo propietario del archivo. Por ejemplo:

```
# mkdir test2.dir/subdir.2
# ls -dv test2.dir/subdir.2
drwxr-xr-x+  2 root     root           2 May 20 15:52 test2.dir/subdir.2
     0:user:gozer:list_directory/read_data/add_file/write_data:file_inherit
         /inherit_only:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

EJEMPLO 8–6 Concesión de herencia de ACL en archivos y directorios *(Continuación)*

En los ejemplos siguientes se identifican las ACL de archivo y directorio que se aplican si se establecen los indicadores `file_inherit` y `dir_inherit`.

En este ejemplo, al usuario `gozer` se le conceden permisos de lectura, escritura y ejecución que se heredan para archivos y directorios recientemente creados:

```
# chmod A+user:gozer:read_data/write_data/execute:file_inherit/dir_inherit:allow
test3.dir
# ls -dv test3.dir
drwxr-xr-x+  2 root     root           2 May 20 15:53 test3.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
         :file_inherit/dir_inherit:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow

# touch test3.dir/file.3
# ls -v test3.dir/file.3
-rw-r--r--+  1 root     root           0 May 20 15:58 test3.dir/file.3
     0:user:gozer:write_data/execute:deny
     1:user:gozer:read_data/write_data/execute:allow
     2:owner@:execute:deny
     3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     4:group@:write_data/append_data/execute:deny
     5:group@:read_data:allow
     6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow

# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+  2 root     root           2 May 20 15:59 test3.dir/subdir.1
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
         :file_inherit/dir_inherit/inherit_only:allow
     1:user:gozer:add_file/write_data:deny
     2:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
     3:owner@::deny
     4:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     5:group@:add_file/write_data/add_subdirectory/append_data:deny
     6:group@:list_directory/read_data/execute:allow
     7:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
```

EJEMPLO 8–6 Concesión de herencia de ACL en archivos y directorios *(Continuación)*

```
     8:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

En estos ejemplos, debido a que los permisos del directorio principal para `group@` y `everyone@` deniegan permisos de lectura y ejecución, al usuario `gozer` se le deniegan permisos de escritura y ejecución. El valor predeterminado de la propiedad `aclinherit` es `restricted`, lo cual significa que no se heredan los permisos `write_data` y `execute`.

En este ejemplo, al usuario `gozer` se le conceden permisos de lectura, escritura y ejecución que se heredan para archivos creados recientemente. Pero no se propagan por el resto del contenido del directorio.

```
# chmod A+user:gozer:read_data/write_data/execute:file_inherit/no_propagate:allow
test4.dir
# ls -dv test4.dir
drwxr-xr-x+  2 root     root           2 May 20 16:02 test4.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
         :file_inherit/no_propagate:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Como puede verse en este ejemplo, si se crea un subdirectorio, los permisos `read_data/write_data/execute` del usuario `gozer` no se propagan al nuevo directorio `sub4.dir`:

```
mkdir test4.dir/sub4.dir
# ls -dv test4.dir/sub4.dir
drwxr-xr-x   2 root     root           2 May 20 16:03 test4.dir/sub4.dir
     0:owner@::deny
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     2:group@:add_file/write_data/add_subdirectory/append_data:deny
     3:group@:list_directory/read_data/execute:allow
     4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Como se observa en el ejemplo siguiente, los permisos `read_data/write_data/execute` del usuario `gozer` para archivos se propagan al archivo que acaba de crearse:

EJEMPLO 8–6 Concesión de herencia de ACL en archivos y directorios *(Continuación)*

```
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+  1 root     root           0 May 20 16:04 test4.dir/file.4
     0:user:gozer:write_data/execute:deny
     1:user:gozer:read_data/write_data/execute:allow
     2:owner@:execute:deny
     3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     4:group@:write_data/append_data/execute:deny
     5:group@:read_data:allow
     6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

EJEMPLO 8–7 Herencia de ACL con la propiedad `aclmode` establecida como `passthrough`

Como muestra el siguiente ejemplo, cuando la propiedad `aclmode` del sistema de archivos `tank/cindys` se configura como `passthrough` , el usuario `gozer` hereda la ACL aplicada al directorio `test4.dir` para el archivo `file.4` recién creado:

```
# zfs set aclmode=passthrough tank/cindys
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+  1 root     root           0 May 20 16:08 test4.dir/file.4
     0:user:gozer:write_data/execute:deny
     1:user:gozer:read_data/write_data/execute:allow
     2:owner@:execute:deny
     3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     4:group@:write_data/append_data/execute:deny
     5:group@:read_data:allow
     6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

En esta salida se observa que la ACL `read_data/write_data/execute:allow:file_inherit/dir_inherit` establecida en el directorio principal, `test4.dir`, se transfiere al usuario `gozer`.

EJEMPLO 8–8 Herencia de ACL con la propiedad `aclmode` establecida como `discard`

Si la propiedad `aclmode` en un sistema de archivo se establece en `discard`, las ACL podrían descartarse si cambian los permisos en un directorio. Por ejemplo:

```
# zfs set aclmode=discard tank/cindys
# chmod A+user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
# ls -dv test5.dir
drwxr-xr-x+  2 root     root           2 May 20 16:09 test5.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
```

EJEMPLO 8–8 Herencia de ACL con la propiedad `aclmode` establecida como `discard` *(Continuación)*

```
        :dir_inherit:allow
     1:owner@::deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     3:group@:add_file/write_data/add_subdirectory/append_data:deny
     4:group@:list_directory/read_data/execute:allow
     5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Si, posteriormente, decide restringir los permisos de un directorio, se prescinde de la ACL no trivial. Por ejemplo:

```
# chmod 744 test5.dir
# ls -dv test5.dir
drwxr--r--   2 root     root           2 May 20 16:09 test5.dir
     0:owner@::deny
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     2:group@:add_file/write_data/add_subdirectory/append_data/execute:deny
     3:group@:list_directory/read_data:allow
     4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /execute/write_attributes/write_acl/write_owner:deny
     5:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
         /synchronize:allow
```

EJEMPLO 8–9 Herencia de ACL con la propiedad `aclinherit` definida en `noallow`

En este ejemplo se establecen dos ACL no triviales con herencia de archivos. Una ACL concede el permiso `read_data` y la otra deniega el permiso `read_data`. Asimismo, el ejemplo muestra la manera de especificar dos entradas de control de acceso en el mismo comando `chmod`.

```
# zfs set aclinherit=noallow tank/cindys
# chmod A+user:gozer:read_data:file_inherit:deny,user:lp:read_data:file_inherit:allow
test6.dir
# ls -dv test6.dir
drwxr-xr-x+  2 root     root           2 May 20 16:11 test6.dir
     0:user:gozer:read_data:file_inherit:deny
     1:user:lp:read_data:file_inherit:allow
     2:owner@::deny
     3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/write_xattr/execute/write_attributes/write_acl
         /write_owner:allow
     4:group@:add_file/write_data/add_subdirectory/append_data:deny
     5:group@:list_directory/read_data/execute:allow
     6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
         /write_attributes/write_acl/write_owner:deny
     7:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

EJEMPLO 8–9 Herencia de ACL con la propiedad `aclinherit` definida en `noallow` *(Continuación)*

Como muestra el ejemplo siguiente, al crear un archivo, se prescinde de la ACL que concede el permiso `read_data`.

```
# touch test6.dir/file.6
# ls -v test6.dir/file.6
-rw-r--r--   1 root     root           0 May 20 16:13 test6.dir/file.6
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
         /write_acl/write_owner:allow
     2:group@:write_data/append_data/execute:deny
     3:group@:read_data:allow
     4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
         /write_acl/write_owner:deny
     5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Establecimiento y visualización de ACL en archivos ZFS en formato compacto

En archivos ZFS puede establecer y visualizar permisos en un formato compacto que utiliza 14 caracteres exclusivos para representar los permisos. Las letras que representan los permisos compactos se detallan en la Tabla 8–2 y la Tabla 8–3.

Puede visualizar listas de ACL en formato compacto para archivos y directorios mediante el comando `ls -V`. Por ejemplo:

```
# ls -V file.1
-rw-r--r--   1 root     root      206663 Jun 17 10:07 file.1
           owner@:--x-----------:------:deny
           owner@:rw-p---A-W-Co-:------:allow
           group@:-wxp----------:------:deny
           group@:r-------------:------:allow
        everyone@:-wxp---A-W-Co-:------:deny
        everyone@:r-----a-R-c--s:------:allow
```

Esta salida ACL compacta se describe de este modo:

`owner@` Se deniegan al propietario permisos de ejecución en el archivo (`x= execute`).

`owner@` El propietario puede leer y modificar el contenido del archivo (`rw=read_data/write_data`, `p= append_data`). Asimismo, puede modificar atributos del archivo como indicaciones de hora, atributos extendidos y ACL (`A=write_xattr`, `W=write_attributes` y `C=write_acl`). Además, el propietario puede modificar la propiedad del archivo (`o=write_owner`).

`group@`	Se deniegan al grupo los permisos de modificación y ejecución del archivo (`write_data`, `p=append_data` y `x=execute`).
`group@`	Se conceden al grupo permisos de ejecución en el archivo (`r= read_data`).
`everyone@`	Se deniega a quien no sea propietario del archivo o miembro del grupo propietario el permiso de ejecución o modificación del contenido del archivo y de cualquiera de sus atributos (`w= write_data`, `x=execute`, `p =append_data`, `A=write_xattr` , `W=write_attributes`, `C= write_acl` y `o=write_owner`).
`everyone@`	Quien no sea propietario del archivo o miembro del grupo propietario del archivo y sus atributos (`r= read_data`, `a=append_data`, `R=read_xattr`, `c=read_acl` y `s=synchronize`). El permiso de acceso `synchronize` no está implementado en la actualidad.

El formato compacto de ACL presenta las ventajas siguientes respecto al formato detallado:

- Los permisos se pueden especificar como argumentos posicionales en el comando `chmod`.
- Los guiones (-), que no identifican permisos, se pueden eliminar. Sólo hace falta especificar las letras necesarias.
- Los indicadores de permisos y de herencia se establecen de la misma manera.

Para obtener información sobre el uso del formato de ACL detallado, consulte "Establecimiento y visualización de ACL en archivos ZFS en formato detallado" en la página 250.

EJEMPLO 8–10 Establecimiento y visualización de las ACL en formato compacto

En el ejemplo siguiente, existe una ACL trivial en `file.1`:

```
# ls -V file.1
-rw-r--r--   1 root     root      206663 Jun 17 10:07 file.1
           owner@:--x-----------:------:deny
           owner@:rw-p---A-W-Co-:------:allow
           group@:-wxp----------:------:deny
           group@:r-------------:------:allow
        everyone@:-wxp---A-W-Co-:------:deny
        everyone@:r-----a-R-c--s:------:allow
```

En el ejemplo siguiente, se agregan permisos `read_data/execute` para el usuario `gozer` en `file.1`:

```
# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+  1 root     root      206663 Jun 17 10:07 file.1
       user:gozer:r-x-----------:------:allow
           owner@:--x-----------:------:deny
           owner@:rw-p---A-W-Co-:------:allow
           group@:-wxp----------:------:deny
           group@:r-------------:------:allow
        everyone@:-wxp---A-W-Co-:------:deny
        everyone@:r-----a-R-c--s:------:allow
```

EJEMPLO 8–10 Establecimiento y visualización de las ACL en formato compacto *(Continuación)*

Una forma alternativa de agregar los mismos permisos para el usuario `gozer` consiste en insertar una ACL nueva en una posición determinada, por ejemplo 4. Así se desplazan hacia abajo las posiciones ya existentes entre la 4 y la 6. Por ejemplo:

```
# chmod A4+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+  1 root     root      206663 Jun 17 10:16 file.1
            owner@:--x-----------:------:deny
            owner@:rw-p---A-W-Co-:------:allow
            group@:-wxp----------:------:deny
            group@:r-------------:------:allow
         user:gozer:r-x-----------:------:allow
          everyone@:-wxp---A-W-Co-:------:deny
          everyone@:r-----a-R-c--s:------:allow
```

En el ejemplo siguiente, al usuario `gozer` se le conceden permisos de lectura, escritura y ejecución que se heredan para archivos y directorios recientemente creados.

```
# chmod A+user:gozer:rwx:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+  2 root     root           2 Jun 17 10:19 dir.2
         user:gozer:rwx-----------:fd----:allow
            owner@:--------------:------:deny
            owner@:rwxp---A-W-Co-:------:allow
            group@:-w-p----------:------:deny
            group@:r-x-----------:------:allow
          everyone@:-w-p---A-W-Co-:------:deny
          everyone@:r-x---a-R-c--s:------:allow
```

También puede cortar y pegar indicadores de herencia y permisos de la salida de `ls -V` en el formato compacto de `chmod`. Por ejemplo, para duplicar los permisos e indicadores de herencia en el directorio `dir.2` del usuario `gozer` para el usuario `cindys` en `dir.2`, copie y pegue los permisos y los indicadores de herencia (`rwx-----------:fd----:allow`) en el comando `chmod` como se indica a continuación:

```
# chmod A+user:cindys:rwx-----------:fd----:allow dir.2
# ls -dV dir.2
drwxr-xr-x+  2 root     root           2 Jun 17 10:19 dir.2
        user:cindys:rwx-----------:fd----:allow
         user:gozer:rwx-----------:fd----:allow
            owner@:--------------:------:deny
            owner@:rwxp---A-W-Co-:------:allow
            group@:-w-p----------:------:deny
            group@:r-x-----------:------:allow
          everyone@:-w-p---A-W-Co-:------:deny
          everyone@:r-x---a-R-c--s:------:allow
```

EJEMPLO 8–11 Herencia de ACL con la propiedad `aclinherit` establecida en `passthrough`

Un sistema de archivos que tiene la propiedad `aclinherit` establecida en `passthrough` hereda todas las entradas ACL que se pueden heredar sin que se les apliquen modificaciones al heredarlas. Si la propiedad se establece en `passthrough`, los archivos se crean con permisos determinados por las entradas de control de acceso que se pueden heredar. Si no existen entradas de control de acceso que se puedan heredar y que afecten a los permisos, los permisos se configurarán de acuerdo con los solicitados desde la aplicación.

Los ejemplos siguientes utilizan sintaxis de ACL compacta para mostrar cómo heredar permisos estableciendo la propiedad `aclinherit` en `passthrough`.

En este ejemplo hay una ACL establecida en el directorio `test1.dir` para la forzar la herencia. La sintaxis crea una entrada ACL `owner@`, `group@` y `everyone@` para cada archivo recién creado. Los directorios que cree heredan una entrada ACL `@owner`, `group@` y `everyone@`. Asimismo, los directorios heredan otras seis entradas de control de acceso que propagan las entradas de control de acceso a los directorios y archivos que se creen.

```
# zfs set aclinherit=passthrough tank/cindys
# pwd
/tank/cindys
# mkdir test1.dir

# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow
test1.dir
# ls -Vd test1.dir
drwxrwx---+  2 root     root           2 Jun 17 10:37 test1.dir
            owner@:rwxpdDaARWcCos:fd----:allow
            group@:rwxp----------:fd----:allow
         everyone@:--------------:fd----:allow
```

En este ejemplo, un archivo recién creado hereda la ACL especificada para heredarse en los archivos recién creados.

```
# cd test1.dir
# touch file.1
# ls -V file.1
-rwxrwx---+  1 root     root           0 Jun 17 10:38 file.1
            owner@:rwxpdDaARWcCos:------:allow
            group@:rwxp----------:------:allow
         everyone@:--------------:------:allow
```

En este ejemplo, un directorio que se cree hereda tanto las entradas que controlan el acceso a este directorio como las entradas de control de acceso para la futura propagación a los elementos secundarios del directorio que se cree.

```
# mkdir subdir.1
# ls -dV subdir.1
drwxrwx---+  2 root     root           2 Jun 17 10:39 subdir.1
            owner@:rwxpdDaARWcCos:fdi---:allow
            owner@:rwxpdDaARWcCos:------:allow
```

EJEMPLO 8–11 Herencia de ACL con la propiedad `aclinherit` establecida en `passthrough` *(Continuación)*

```
            group@:rwxp----------:fdi---:allow
            group@:rwxp----------:------:allow
         everyone@:--------------:fdi---:allow
         everyone@:--------------:------:allow
```

Las entradas `-di--` y `f-i---` propagan la herencia y no se tienen en cuenta durante el control de acceso. En este ejemplo, se crea un archivo con una ACL trivial en otro directorio en el que no haya entradas de control de acceso heredadas.

```
# cd /tank/cindys
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r--   1 root     root           0 Jun 17 10:40 file.2
            owner@:--x-----------:------:deny
            owner@:rw-p---A-W-Co-:------:allow
            group@:-wxp----------:------:deny
            group@:r-------------:------:allow
         everyone@:-wxp---A-W-Co-:------:deny
         everyone@:r-----a-R-c--s:------:allow
```

EJEMPLO 8–12 Herencia de ACL con la propiedad `aclinherit` establecida en `passthrough-x`

Cuando la propiedad `aclinherit` está establecida en `passthrough-x` , los archivos se crean con el permiso de ejecución (`x`) para `owner@`, `group@` o `everyone@`, pero sólo si el permiso de ejecución se establece en modo de creación de archivos y una ACE heredable que afecta al modo.

El siguiente ejemplo muestra cómo se heredan los permisos de ejecución estableciendo la propiedad `aclinherit` en `passthrough-x`.

```
# zfs set aclinherit=passthrough-x tank/cindys
```

La siguiente ACL se ha establecido en `/tank/cindys/test1.dir` para proporcionar herencia de ACL ejecutable para los archivos de `owner@`, `group@` y `everyone@`.

```
# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow test1.dir
# ls -Vd test1.dir
drwxrwx---+  2 root     root           2 Jun 17 10:41 test1.dir
            owner@:rwxpdDaARWcCos:fd----:allow
            group@:rwxp----------:fd----:allow
         everyone@:--------------:fd----:allow
```

Un archivo (`file1`) se crea con permisos solicitados `0666`. Los permisos resultantes son `0660`. El permiso de ejecución no se ha heredado porque el modo de creación no lo solicitó.

EJEMPLO 8–12 Herencia de ACL con la propiedad `aclinherit` establecida en `passthrough-x` *(Continuación)*

```
# touch test1.dir/file1
# ls -V test1.dir/file1
-rw-rw----+  1 root     root           0 Jun 17 10:42 test1.dir/file1
            owner@:rw-pdDaARWcCos:------:allow
            group@:rw-p----------:------:allow
         everyone@:--------------:------:allow
```

A continuación, se generará un ejecutable llamado `t` mediante el compilador `cc` en el directorio `testdir`.

```
# cc -o t t.c
# ls -V t
-rwxrwx---+  1 root     root        7396 Jun 17 10:50 t
               owner@:rwxpdDaARWcCos:------:allow
               group@:rwxp----------:------:allow
            everyone@:--------------:------:allow
```

Los permisos resultantes son `0770` porque `cc` solicitó permisos `0777`, que provocaron que el permiso de ejecución se heredara de las entradas `owner@`, `group@` y `everyone@`.

◆ ◆ ◆ CAPÍTULO 9

Administración delegada de ZFS

Este capítulo describe la forma de utilizar la administración delegada para permitir que los usuarios sin privilegios puedan efectuar tareas de administración de ZFS.

Descripción general de la administración delegada de ZFS

Esta función permite distribuir permisos más concretos a determinados usuarios, grupos o a todo el mundo. Se permiten dos tipos de permisos delegados:

- Se pueden especificar permisos concretos a determinadas personas, por ejemplo crear, destruir, montar, crear instantánea, etcétera.
- Se pueden definir grupos de permisos denominados *conjuntos de permisos*. Un conjunto de permisos se puede actualizar posteriormente y todos los usuarios de dicho conjunto adquieren ese cambio de forma automática. Los conjuntos de permisos comienzan con el carácter `@` y tienen un límite de 64 caracteres. Después del carácter `@`, los demás caracteres del nombre del conjunto están sujetos a las mismas restricciones que los nombres de sistemas de archivos ZFS normales.

La administración delegada de ZFS proporciona funciones parecidas al modelo de seguridad RBAC. Esta función aporta las ventajas siguientes en la administración de agrupaciones de almacenamiento y sistemas de archivos ZFS:

- Si se migra la agrupación, se mantienen los permisos en la agrupación de almacenamiento ZFS.
- Proporciona herencia dinámica para poder controlar cómo se propagan los permisos por los sistemas de archivos.

- Se puede configurar para que sólo el creador de un sistema de archivos pueda destruir el sistema de archivos.
- Se pueden distribuir permisos en determinados sistemas de archivos. Los sistemas de archivos creados pueden designar permisos automáticamente.
- Proporciona administración NFS simple. Por ejemplo, un usuario que cuenta con permisos explícitos puede crear una instantánea por NFS en el correspondiente directorio `.zfs/snapshot`.

A la hora de distribuir tareas de ZFS, plantéese la posibilidad de utilizar la administración delegada. Si desea información sobre el uso de RBAC para llevar a cabo tareas generales de administración de Solaris, consulte la Parte III, "Roles, Rights Profiles, and Privileges" de *System Administration Guide: Security Services*.

Inhabilitación de permisos delegados de ZFS

Puede controlar las funciones de administración delegada mediante la propiedad `delegation` de la agrupación. Por ejemplo:

```
# zpool get delegation users
NAME  PROPERTY    VALUE       SOURCE
users  delegation  on          default
# zpool set delegation=off users
# zpool get delegation users
NAME  PROPERTY    VALUE       SOURCE
users  delegation  off         local
```

De forma predeterminada se activa la propiedad `delegation`.

Delegación de permisos de ZFS

Utilice el comando `zfs allow` para conceder permisos en conjuntos de datos ZFS a usuarios que no sean root de las maneras siguientes:

- Los permisos individuales se pueden conceder a un grupo, un usuario o a todo el mundo.
- Los grupos de permisos individuales se pueden conceder como *conjunto de permisos* a un grupo, un usuario o a todo el mundo.
- Los permisos se pueden conceder localmente sólo al conjunto de datos actual o a todos los elementos descendientes de dicho conjunto de datos.

En la tabla siguiente figuran las operaciones que se pueden delegar y los permisos dependientes que se necesitan para efectuar las operaciones delegadas.

Permiso (subcomando)	Descripción	Dependencias
`allow`	Capacidad para conceder a otro usuario los permisos que tiene uno mismo.	También debe tener el permiso que se está concediendo.
`clone`	Capacidad para clonar cualquier instantánea del conjunto de datos.	También se debe disponer de las capacidades `create` y `mount` en el sistema de archivos original.
`create`	Capacidad para crear conjuntos de archivos descendientes.	También se debe disponer de la capacidad `mount`.
`destroy`	Capacidad para destruir un conjunto de datos.	También se debe disponer de la capacidad `mount`.
`mount`	Capacidad para montar y desmontar un conjunto de datos, así como para crear y destruir vínculos de dispositivos de volumen.	
`promote`	Capacidad para promover un clon a un conjunto de datos.	También se debe disponer de las capacidades `mount` y `promote` en el sistema de archivos original.
`receive`	Capacidad para crear sistemas de archivos descendientes mediante el comando `zfs receive`.	También se debe disponer de las capacidades `mount` y `create`.
`rename`	Capacidad para cambiar el nombre a un conjunto de datos.	También se debe disponer de las capacidades `create` y `mount` en el nuevo elemento principal.
`rollback`	Capacidad para invertir una instantánea.	
`send`	Capacidad para enviar un flujo de instantáneas.	
`share`	Capacidad para compartir y anular la compartición de un conjunto de datos.	
`snapshot`	Capacidad para obtener una instantánea de un conjunto de datos.	

Puede delegar el siguiente conjunto de permisos, pero el permiso puede estar limitado a acceso, lectura o cambio:

- `groupquota`
- `groupused`
- `userprop`
- `userquota`
- `userused`

Además, puede delegar las siguientes propiedades de ZFS a usuarios que no sean root:

- aclinherit
- aclmode
- atime
- canmount
- casesensitivity
- checksum
- compression
- copies
- devices
- exec
- mountpoint
- nbmand
- normalization
- primarycache
- quota
- readonly
- recordsize
- refreservation
- reservation
- secondarycache
- setuid
- shareiscsi
- sharenfs
- sharesmb
- snapdir
- utf8only
- version
- volblocksize
- volsize
- vscan
- xattr
- zoned

Algunas de estas propiedades sólo se pueden establecer al crear el conjunto de datos. Para obtener una descripción de estas propiedades, consulte "Introducción a las propiedades de ZFS" en la página 189.

Delegación de permisos de ZFS (`zfs allow`)

La sintaxis `zfs allow` es la siguiente:

```
zfs allow -[ldugecs] everyone|user|group[,...] perm|@setname,...] filesystem| volume
```

La siguiente sintaxis de `zfs allow` (en negrita) identifica a quién se delegan los permisos:

`zfs allow [-uge]|user|group|everyone` `[,...]` *filesystem* | *volume*

Se pueden especificar varias entidades en una lista separada por comas. Si no se especifican opciones de `-uge`, el argumento se interpreta de forma preferente como la palabra clave `everyone`, después como nombre de usuario y, en último lugar, como grupo de nombre. Para especificar un usuario o grupo denominado "everyone", utilice la opción `-u` o `-g`. Para especificar un grupo con el mismo nombre que un usuario, utilice la opción `-g`. La opción `-c` concede permisos de create-time.

La siguiente sintaxis de `zfs allow` (en negrita) identifica cómo se especifican los permisos y conjuntos de permisos:

`zfs allow [-s] ... perm|@setname [,...]` *filesystem* | *volume*

Se pueden especificar varios permisos en una lista separada por comas. Los nombres de permisos son los mismos que las propiedades y los subcomandos de ZFS. Para obtener más información, consulte la sección anterior.

Los permisos se pueden agregar a *conjuntos de permisos* y los identifica la opción `-s`. Otros comandos de `zfs allow` pueden utilizar conjuntos de permisos para el sistema de archivos especificado y sus elementos descendientes. Los conjuntos de permisos se evalúan dinámicamente, por lo tanto los cambios que haya en un conjunto se actualizan de manera inmediata. Los conjuntos de permisos siguen las mismas convenciones de denominación que los sistemas de archivos ZFS; sin embargo, el nombre debe comenzar con el signo de arroba (`@`) y tener un máximo de 64 caracteres.

La siguiente sintaxis de `zfs allow` (en negrita) identifica cómo se delegan los permisos:

`zfs allow [-ld]` `...` `...` *filesystem* | *volume*

La opción `-l` indica que el permiso se concede para el conjunto de datos especificado pero no a los elementos descendientes, a menos que también se especifique la opción `-d`. La opción `-d` indica que el permiso se concede a los conjuntos de datos descendientes y no a este conjunto de datos, a menos que también se especifique la opción `-l`. Si no se especifica ninguna de las opciones de `-ld`, los permisos se conceden para el volumen o sistema de archivos y todos sus elementos descendientes.

Eliminación de permisos delegados de ZFS (`zfs unallow`)

Mediante el comando `zfs unallow` se pueden eliminar los permisos que se han concedido.

Por ejemplo, suponga que ha delegado los permisos `create`, `destroy`, `mount` y `snapshot` de la forma siguiente:

```
# zfs allow cindys create,destroy,mount,snapshot tank/cindys
# zfs allow tank/cindys
-------------------------------------------------------------
Local+Descendent permissions on (tank/cindys)
        user cindys create,destroy,mount,snapshot
-------------------------------------------------------------
```

Para eliminar estos permisos, debe utilizar una sintaxis similar a la siguiente:

```
# zfs unallow cindys tank/cindys
# zfs allow tank/cindys
```

Uso de la administración delegada de ZFS

En esta sección se proporcionan ejemplos de delegación y visualización de permisos de ZFS.

Delegación de permisos ZFS (ejemplos)

EJEMPLO 9–1 Delegación de permisos a un determinado usuario

Si concede los permisos `create` y `mount` a un determinado usuario, compruebe que dicho usuario disponga de permisos en el punto de montaje subyacente.

Por ejemplo, para proporcionar al usuario los permisos `marks` `create` y `mount` en`tank`, primero establezca los permisos:

```
# chmod A+user:marks:add_subdirectory:fd:allow /tank
```

A continuación, utilice el comando `zfs allow` para conceder los permisos `create` , `destroy` y `mount`. Por ejemplo:

```
# zfs allow marks create,destroy,mount tank
```

El usuario `marks` ya puede crear sus propios sistemas de archivos en el sistema de archivos `tank`. Por ejemplo:

```
# su marks
marks$ zfs create tank/marks
marks$ ^D
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied
```

EJEMPLO 9–2 Delegación de los permisos create y destroy en un grupo

El ejemplo siguiente muestra cómo configurar un sistema de archivos de forma que cualquier integrante del grupo `staff` pueda crear y montar sistemas de archivos en el sistema de archivos `tank`, así como destruir sus propios sistemas de archivos. Ahora bien, los miembros del grupo `staff` no pueden destruir los sistemas de archivos de nadie más.

EJEMPLO 9–2 Delegación de los permisos create y destroy en un grupo *(Continuación)*

```
# zfs allow staff create,mount tank
# zfs allow -c create,destroy tank
# zfs allow tank
-------------------------------------------------------------
Create time permissions on (tank)
        create,destroy
Local+Descendent permissions on (tank)
        group staff create,mount
-------------------------------------------------------------
# su cindys
cindys% zfs create tank/cindys
cindys% exit
# su marks
marks% zfs create tank/marks/data
marks% exit
cindys% zfs destroy tank/marks/data
cannot destroy 'tank/mark': permission denied
```

EJEMPLO 9–3 Delegación de permisos en el nivel correcto del sistema de archivos

Compruebe que conceda permisos a los usuarios en el nivel correcto del sistema de archivos. Por ejemplo, al usuario `marks` se le concede los permisos `create`, `destroy` y `mount` para los sistemas de archivos local y descendiente. Al usuario `marks` se le concede permiso local para crear una instantánea del sistema de archivos `tank`, pero no puede crear una instantánea de su propio sistema de archivos. Así pues, no se le ha concedido el permiso `snapshot` en el nivel correcto del sistema de archivos.

```
# zfs allow -l marks snapshot tank
# zfs allow tank
-------------------------------------------------------------
Local permissions on (tank)
        user marks snapshot
Local+Descendent permissions on (tank)
        user marks create,destroy,mount
-------------------------------------------------------------
# su marks
marks$ zfs snapshot tank/@snap1
marks$ zfs snapshot tank/marks@snap1
cannot create snapshot 'mark/marks@snap1': permission denied
```

Para conceder permiso al usuario `marks` en el nivel descendiente, utilice la opción `zfs allow -d`. Por ejemplo:

```
# zfs unallow -l marks snapshot tank
# zfs allow -d marks snapshot tank
# zfs allow tank
-------------------------------------------------------------
Descendent permissions on (tank)
        user marks snapshot
Local+Descendent permissions on (tank)
        user marks create,destroy,mount
-------------------------------------------------------------
```

EJEMPLO 9–3 Delegación de permisos en el nivel correcto del sistema de archivos *(Continuación)*

```
# su marks
$ zfs snapshot tank@snap2
cannot create snapshot 'tank@snap2': permission denied
$ zfs snapshot tank/marks@snappy
```

El usuario `marks` ahora sólo puede crear una instantánea por debajo del nivel de `tank`.

EJEMPLO 9–4 Definición y uso de permisos delegados complejos

Puede conceder permisos a usuarios o grupos. Por ejemplo, el siguiente comando `zfs allow` concede determinados permisos al grupo `staff`. Asimismo, se conceden los permisos `destroy` y `snapshot` una vez creados los sistemas de archivos de `tank`.

```
# zfs allow staff create,mount tank
# zfs allow -c destroy,snapshot tank
# zfs allow tank
-------------------------------------------------------------
Create time permissions on (tank)
        destroy,snapshot
Local+Descendent permissions on (tank)
        group staff create,mount
-------------------------------------------------------------
```

Debido a que el usuario `marks` pertenece al grupo `staff`, puede crear sistemas de archivos en `tank`. Además, el usuario `marks` puede crear una instantánea de `tank/marks2` porque dispone de los correspondientes permisos para hacerlo. Por ejemplo:

```
# su marks
$ zfs create tank/marks2
$ zfs allow tank/marks2
-------------------------------------------------------------
Local permissions on (tank/marks2)
        user marks destroy,snapshot
-------------------------------------------------------------
Create time permissions on (tank)
        destroy,snapshot
Local+Descendent permissions on (tank)
        group staff create
        everyone mount
-------------------------------------------------------------
```

Sin embargo, no puede crear una instantánea de `tank/marks` porque carece de los correspondientes permisos. Por ejemplo:

```
$ zfs snapshot tank/marks2@snap1
$ zfs snapshot tank/marks@snappp
cannot create snapshot 'tank/marks@snappp': permission denied
```

Si dispone del permiso `create` en su directorio principal, puede crear directorios de instantáneas. Esta situación hipotética es útil si el sistema de archivos están montado por NFS. Por ejemplo:

EJEMPLO 9–4 Definición y uso de permisos delegados complejos *(Continuación)*

```
$ cd /tank/marks2
$ ls
$ cd .zfs
$ ls
snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x   2 marks    staff          2 Dec 15 13:53 snap1
$ pwd
/tank/marks2/.zfs/snapshot
$ mkdir snap2
$ zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
tank                   264K  33.2G  33.5K  /tank
tank/marks            24.5K  33.2G  24.5K  /tank/marks
tank/marks2             46K  33.2G  24.5K  /tank/marks2
tank/marks2@snap1     21.5K      -  24.5K  -
tank/marks2@snap2         0      -  24.5K  -
$ ls
snap1  snap2
$ rmdir snap2
$ ls
snap1
```

EJEMPLO 9–5 Definición y uso de un conjunto de permisos delegados de ZFS

En el ejemplo siguiente se muestra cómo crear un conjunto de permisos `@myset`, conceder el grupo de permisos y cambiar el nombre del permiso para el grupo `staff` respecto al sistema de archivos `tank`. `cindys`, usuario del grupo `staff`, tiene permiso para crear un sistema de archivos en `tank`. Sin embargo, el usuario `lp` no tiene permiso para crear un sistema de archivos en `tank`.

```
# zfs allow -s @myset create,destroy,mount,snapshot,promote,clone,readonly tank
# zfs allow tank
-------------------------------------------------------------
Permission sets on (tank)
        @myset clone,create,destroy,mount,promote,readonly,snapshot
-------------------------------------------------------------
# zfs allow staff @myset,rename tank
# zfs allow tank
-------------------------------------------------------------
Permission sets on (tank)
        @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions on (tank)
        group staff @myset,rename
# chmod A+group:staff:add_subdirectory:fd:allow tank
# su cindys
cindys% zfs create tank/data
Cindys% zfs allow tank
-------------------------------------------------------------
Permission sets on (tank)
        @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions on (tank)
```

EJEMPLO 9–5 Definición y uso de un conjunto de permisos delegados de ZFS *(Continuación)*

```
        group staff @myset,rename
------------------------------------------------------------
cindys% ls -l /tank
total 15
drwxr-xr-x   2 cindys   staff          2 Aug  8 14:10 data
cindys% exit
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied
```

Visualización de permisos delegados de ZFS

Para visualizar permisos puede utilizar el comando siguiente:

```
# zfs allow dataset
```

Este comando muestra los permisos que se establecen o permiten en este conjunto de datos. La salida contiene los componentes siguientes:

- Conjuntos de permisos
- Permisos específicos o permisos create-time
- Conjunto de datos local
- Conjuntos de datos locales y descendientes
- Sólo conjuntos de datos descendientes

EJEMPLO 9–6 Visualización de permisos de administración delegados básicos

La salida de este ejemplo indica que el usuario `cindys` tiene permisos para crear, destruir, montar y crear instantáneas en el sistema de archivos `tank/cindys`.

```
# zfs allow tank/cindys
       -------------------------------------------------------------
       Local+Descendent permissions on (tank/cindys)
               user cindys create,destroy,mount,snapshot
```

EJEMPLO 9–7 Visualización de permisos de administración delegados complejos

La salida de este ejemplo indica los permisos siguientes en los sistemas de archivos `pool/fred` y `pool`.

Para el sistema de archivos `pool/fred`:

- Se definen dos conjuntos de permisos:
  - `@eng` (create, destroy, snapshot, mount, clone, promote, rename)
  - `@simple` (create, mount)

EJEMPLO 9–7 Visualización de permisos de administración delegados complejos *(Continuación)*

- Los permisos create-time se establecen para el conjunto de permisos `@eng` y la propiedad `mountpoint`. Create-time significa que, tras la creación de un conjunto de permisos, se conceden el conjunto de permisos `@eng` y la propiedad `mountpoint`.
- Al usuario`tom` se le concede el conjunto de permisos `@eng`; al usuario `joe` se le conceden los permisos `create`, `destroy` y `mount` para sistemas de archivos locales.
- Al usuario `fred` se le concede el conjunto de permisos `@basic`, así como los permisos `share` y `rename` para los sistemas de archivos locales y descendientes.
- Al usuario `barney` y al grupo de usuarios `staff` se les concede el grupo de permisos `@basic` sólo para sistemas de archivos descendientes.

Para el sistema de archivos `pool`:

- Se define el conjunto de permisos `@simple` (create, destroy, mount).
- Al grupo `staff` se le concede el conjunto de permisos `@simple` en el sistema de archivos local.

A continuación se muestra el resultado de este ejemplo:

```
$ zfs allow pool/fred
-------------------------------------------------------------------------------
Permission sets on (pool/fred)
        @eng create,destroy,snapshot,mount,clone,promote,rename
        @simple create,mount
Create time permissions on (pool/fred)
        @eng,mountpoint
Local permissions on (pool/fred)
        user tom @eng
        user joe create,destroy,mount
Local+Descendent permissions on (pool/fred)
        user fred @basic,share,rename
Descendent permissions on (pool/fred)
        user barney @basic
        group staff @basic
-------------------------------------------------------------------------------
Permission sets on (pool)
        @simple create,destroy,mount
Local permissions on (pool)
        group staff @simple
-------------------------------------------------------------------------------
```

Eliminación de permisos ZFS (ejemplos)

El comando `zfs unallow` se usa para eliminar permisos concedidos. Por ejemplo, el usuario `cindys` tiene permisos para crear, destruir, montar y crear instantáneas en el sistema de archivos `tank/cindys`.

```
# zfs allow cindys create,destroy,mount,snapshot tank/cindys
       # zfs allow tank/cindys
       -------------------------------------------------------------
       Local+Descendent permissions on (tank/cindys)
               user cindys create,destroy,mount,snapshot
       -------------------------------------------------------------
```

La siguiente sintaxis de `zfs unallow` elimina el permiso para crear instantáneas del usuario `cindys` en el sistema de archivos `tank/cindys`:

```
# zfs unallow cindys snapshot tank/cindys
# zfs allow tank/cindys
-------------------------------------------------------------
Local+Descendent permissions on (tank/cindys)
        user cindys create,destroy,mount
-------------------------------------------------------------
cindys% zfs create tank/cindys/data
cindys% zfs snapshot tank/cindys@today
cannot create snapshot 'tank/cindys@today': permission denied
```

Otro ejemplo: el usuario `marks` tiene los permisos siguientes en el sistema de archivos `tank/marks`:

```
# zfs allow tank/marks
-------------------------------------------------------------
Local+Descendent permissions on (tank/marks)
        user marks create,destroy,mount
-------------------------------------------------------------
```

En este ejemplo, la siguiente sintaxis de `zfs unallow` elimina todos los permisos del usuario `marks` de `tank/marks`:

```
# zfs unallow marks tank/marks
```

La siguiente sintaxis de `zfs unallow` elimina un conjunto de permisos del sistema de archivos `tank`.

```
# zfs allow tank
-------------------------------------------------------------
Permission sets on (tank)
        @myset clone,create,destroy,mount,promote,readonly,snapshot
Create time permissions on (tank)
        create,destroy,mount
Local+Descendent permissions on (tank)
        group staff create,mount
-------------------------------------------------------------
# zfs unallow -s @myset tank
$ zfs allow tank
-------------------------------------------------------------
Create time permissions on (tank)
        create,destroy,mount
Local+Descendent permissions on (tank)
        group staff create,mount
-------------------------------------------------------------
```

◆ ◆ ◆ CAPÍTULO 10

Temas avanzados de Oracle Solaris ZFS

En este capítulo se describen los volúmenes de ZFS, el uso de ZFS en un sistema Solaris con zonas instaladas, agrupaciones raíz alternativas de ZFS y perfiles de derechos de ZFS.

Este capítulo se divide en las secciones siguientes:

Volúmenes de ZFS

Un volumen ZFS es un conjunto de datos que representa un dispositivo de bloques. Los volúmenes ZFS se identifican como dispositivos en el directorio `/dev/zvol/{dsk,rdsk}/pool`.

En el ejemplo siguiente, se crea un volumen de ZFS de 5 GB, `tank/vol`:

```
# zfs create -V 5gb tank/vol
```

Al crear un volumen, automáticamente se reserva espacio para el tamaño inicial del volumen, a fin de evitar imprevistos. Por ejemplo, si disminuye el tamaño del volumen, los datos podrían dañarse. El cambio del volumen se debe realizar con mucho cuidado.

Además, si crea una instantánea de un volumen que cambia de tamaño, podría provocar incoherencias en el sistema de archivos al intentar restaurar una versión anterior de la instantánea o crear un clon de ésta.

Para obtener información sobre las propiedades de sistemas de archivos que se pueden aplicar a volúmenes, consulte la Tabla 6–1.

Si utiliza un sistema Solaris con zonas instaladas, los volúmenes de ZFS no se pueden crear ni clonar en una zona no global. Cualquier intento de hacerlo, fallará. Para obtener información sobre el uso de volúmenes de ZFS en una zona global, consulte "Adición de volúmenes de ZFS a una zona no global" en la página 286.

Uso de un volumen de ZFS como dispositivo de volcado o intercambio

Durante la instalación de un sistema de archivos raíz ZFS o una migración desde un sistema de archivos raíz UFS, se crea un dispositivo de intercambio en un volumen ZFS de la agrupación raíz ZFS. Por ejemplo:

```
# swap -l
swapfile                  dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap 253,3        16  8257520  8257520
```

Durante la instalación de un sistema de archivos raíz ZFS o una migración desde un sistema de archivos raíz UFS, se crea un dispositivo de volcado en un volumen ZFS de la agrupación raíz ZFS. Después de configurarse, no hace falta administrar el dispositivo de volcado. Por ejemplo:

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
  Savecore enabled: yes
```

Si necesita modificar el área de intercambio o el dispositivo de volcado tras instalar o actualizar el sistema, utilice los comandos `swap` y `dumpadm` como en las versiones anteriores de Solaris. Para crear un área de intercambio adicional, cree un volumen ZFS de un tamaño específico y permita el intercambio en dicho dispositivo. Por ejemplo:

```
# zfs create -V 2G rpool/swap2
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile                   dev  swaplo blocks   free
/dev/zvol/dsk/rpool/swap  256,1      16 2097136 2097136
/dev/zvol/dsk/rpool/swap2 256,5      16 4194288 4194288
```

No intercambie a un archivo en un sistema de archivos ZFS. La configuración de archivos de intercambio ZFS no es posible.

Para obtener información sobre cómo ajustar el tamaño de los volúmenes de intercambio y volcado, consulte "Ajuste del tamaño de los dispositivos de intercambio y volcado ZFS" en la página 168.

Uso de un volumen de ZFS como objetivo iSCSI de Solaris

Puede crear fácilmente un volumen de ZFS como objetivo iSCSI estableciendo la propiedad `shareiscsi` en el volumen. Por ejemplo:

```
# zfs create -V 2g tank/volumes/v2
# zfs set shareiscsi=on tank/volumes/v2
# iscsitadm list target
Target: tank/volumes/v2
    iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
    Connections: 0
```

Tras crear el objetivo iSCSI, configure el iniciador iSCSI. Para obtener más información sobre objetivos e iniciadores iSCSI de Solaris, consulte el Capítulo 14, "Configuring Oracle Solaris iSCSI Targets and Initiators (Tasks)" de *System Administration Guide: Devices and File Systems*.

Nota – Los objetivos iSCSI de Solaris también se pueden crear y administrar con el comando `iscsitadm`. Si se establece la propiedad `shareiscsi` en un volumen de ZFS, no utilice el comando `iscsitadm` para crear el mismo dispositivo de destino. De lo contrario, se creará información de destino duplicada para el mismo dispositivo.

Un volumen de ZFS como objetivo iSCSI se administra de la misma manera que cualquier otro conjunto de datos de ZFS. Sin embargo, las funciones `rename`, `export` e `import` son algo distintas en los objetivos iSCSI.

- Si se cambia el nombre de un volumen de ZFS, el objetivo iSCSI se sigue llamando de la misma forma. Por ejemplo:

  ```
  # zfs rename tank/volumes/v2 tank/volumes/v1
  # iscsitadm list target
  Target: tank/volumes/v1
      iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
      Connections: 0
  ```

- La exportación de una agrupación que contenga un volumen de ZFS compartido elimina el objetivo. La importación de una agrupación que contenga un volumen de ZFS compartido hace que se comparta el objetivo. Por ejemplo:

  ```
  # zpool export tank
  # iscsitadm list target
  # zpool import tank
  # iscsitadm list target
  Target: tank/volumes/v1
      iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
      Connections: 0
  ```

Toda la información de configuración de objetivos iSCSI se guarda con el conjunto de datos. Al igual que un sistema de archivos NFS compartido, un objetivo iSCSI que se importa a otro sistema se comparte correspondientemente.

Uso de ZFS en un sistema Solaris con zonas instaladas

Las secciones siguientes explican cómo utilizar ZFS en un sistema con zonas de Oracle Solaris:

Para obtener información sobre cómo configurar zonas en un sistema con un sistema de archivos raíz ZFS que se va a migrar o al que se aplicarán parches con Actualización automática de Solaris, consulte "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (Solaris 10 10/08)" en la página 151 o "Uso de Actualización automática de Oracle Solaris para migrar o actualizar un sistema con zonas (al menos Solaris 10 5/09)" en la página 156.

Al asociar conjuntos de datos de ZFS con zonas, hay que tener en cuenta los puntos siguientes:

- Puede agregar un sistema de archivos o un clon de ZFS a una zona no global con o sin delegación de control administrativo.
- Puede agregar un volumen de ZFS como dispositivo a zonas no globales.
- Por ahora no es posible asociar instantáneas de ZFS con zonas.

En las secciones siguientes, un conjunto de datos de ZFS hace referencia a un sistema de archivos o un clon.

La adición de un conjunto de datos permite que la zona no global comparta espacio en el disco con la zona global, si bien el administrador de zona no puede controlar las propiedades ni crear sistemas de archivos en la jerarquía de sistemas de archivos subyacente. Es lo mismo que agregar cualquier otro sistema de archivos a una zona; es aconsejable utilizarlo si la finalidad principal es compartir espacio.

ZFS permite también la delegación de conjuntos de datos a una zona no global, con lo cual el administrador de zona dispone de control absoluto sobre el conjunto de datos y todos sus conjuntos de datos secundarios. El administrador de zona puede crear y destruir sistemas de archivos o clones de ese conjunto de datos, así como modificar las propiedades de los conjuntos de datos. El administrador de zona no puede incidir en los conjuntos de datos que no se hayan agregado a la zona ni sobrepasar las cuotas de nivel superior establecidas en el conjunto de datos delegado.

Al utilizar ZFS en un sistema con zonas Oracle Solaris instaladas hay que tener en cuenta los puntos siguientes:

- Un sistema de archivos ZFS agregado a una zona no global debe tener la propiedad `mountpoint` establecida en `legacy`.

- Debido al CR 6449301, no agregue ningún conjunto de datos ZFS a una zona no global cuando configure la zona no global. En lugar de ello, agregue un conjunto de datos ZFS tras la instalación de la zona.
- Cuando tanto un origen `zonepath` como un destino `zonepath` residen en un sistema de archivos ZFS y se encuentran en la misma agrupación, `zoneadm clone` utiliza automáticamente el clon de ZFS para clonar una zona. El comando `zoneadm clone` crea una instantánea de ZFS de la `zonepath` de origen y configura la `zonepath` de destino. No puede utilizar el comando `zfs clone` para clonar una zona. Si desea más información, consulte la Parte II, "Zonas" de *Guía de administración de sistemas: administración de recursos y contenedores de Oracle Solaris y zonas de Oracle Solaris*.
- Si delega un sistema de archivos ZFS a una zona no global, debe eliminar ese sistema de archivos de la zona no global antes de utilizar Actualización automática de Oracle Solaris. De lo contrario, la Actualización automática de Oracle fallará debido a un error del sistema de archivos de sólo lectura.

Adición de sistemas de archivos ZFS a una zona no global

Un sistema de archivos ZFS se puede agregar como sistema de archivos genérico si la finalidad es compartir espacio con la zona global. Un sistema de archivos ZFS agregado a una zona no global debe tener la propiedad `mountpoint` establecida en `legacy`.

Un sistema de archivos ZFS puede agregarse a una zona no global mediante el comando `zonecfg` y el subcomando `add fs`.

En el ejemplo siguiente, un administrador de zona global agrega a la zona no global un sistema de archivos ZFS desde la zona global:

```
# zonecfg -z zion
zonecfg:zion> add fs
zonecfg:zion:fs> set type=zfs
zonecfg:zion:fs> set special=tank/zone/zion
zonecfg:zion:fs> set dir=/export/shared
zonecfg:zion:fs> end
```

Esta sintaxis agrega el sistema de archivos ZFS `tank/zone/zion` a la zona ya configurada `zion`, montada en `/export/shared`. La propiedad `mountpoint` del sistema de archivos se debe establecer en `legacy` y el sistema de archivos ya no se puede montar en otra ubicación. El administrador de zona puede crear y destruir archivos en el sistema de archivos. El sistema de archivos no se puede volver a montar en una ubicación distinta; el administrador de zona tampoco puede modificar propiedades del sistema de archivos, por ejemplo `atime`, `readonly` o `compression`. El administrador de zona global se encarga de configurar y controlar las propiedades del sistema de archivos.

Para más información sobre el comando `zonecfg` y la configuración de tipos de recursos con `zonecfg`, consulte la Parte II, "Zonas" de *Guía de administración de sistemas: administración de recursos y contenedores de Oracle Solaris y zonas de Oracle Solaris*.

Delegación de conjuntos de datos a una zona no global

Para cumplir el objetivo principal, que es delegar la administración del almacenamiento a una zona, ZFS permite agregar conjuntos de datos a una zona no global mediante el comando `zonecfg` y el subcomando `add dataset`.

En el ejemplo siguiente, un administrador de zona global delega a la zona no global un sistema de archivos ZFS desde la zona global:

```
# zonecfg -z zion
zonecfg:zion> add dataset
zonecfg:zion:dataset> set name=tank/zone/zion
zonecfg:zion:dataset> end
```

A diferencia de agregar un sistema de archivos, esta sintaxis hace que el sistema de archivos ZFS `tank/zone/zion` quede visible en la zona ya configurada `zion`. El administrador de zona puede establecer las propiedades del sistema de archivos, así como crear sistemas de archivos descendientes. Además, puede crear instantáneas y clones, y controlar toda la jerarquía del sistema de archivos.

Si utiliza Actualización automática Oracle de Solaris para actualizar el entorno de inicio de ZFS con zonas no globales, suprima en primer lugar cualquier conjunto de datos delegado. De lo contrario, Actualización automática Oracle de Solaris fallará por un error de sistema de archivos de sólo lectura. Por ejemplo:

```
zonecfg:zion>
zonecfg:zion> remove dataset name=tank/zone/zion
zonecfg:zion1> exit
```

Para obtener más información sobre las acciones factibles en las zonas, consulte "Administración de propiedades de ZFS en una zona" en la página 287.

Adición de volúmenes de ZFS a una zona no global

Los volúmenes de ZFS no se pueden agregar a una zona no global mediante el comando `zonecfg` y el subcomando `add dataset`. Sin embargo, pueden agregarse volúmenes a una zona utilizando el comando `zonecfg` y el subcomando `add device`.

En el ejemplo siguiente, un administrador de zona global agrega a la zona no global un volumen ZFS desde la zona global:

```
# zonecfg -z zion
zion: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zion> create
zonecfg:zion> add device
zonecfg:zion:device> set match=/dev/zvol/dsk/tank/vol
zonecfg:zion:device> end
```

Esta sintaxis añade el volumen `tank/vol` a la zona `zion`. Agregar un volumen sin formato a una zona conlleva riesgos en la seguridad, incluso si el volumen no se corresponde con un dispositivo físico. En particular, el administrador de zona podría crear sistemas de archivos incorrectamente formados que causen confusión en el sistema al intentar un montaje. Para obtener más información sobre cómo agregar dispositivos a zonas y sus riesgos en la seguridad, consulte "Interpretación de la propiedad zoned" en la página 288.

Para obtener más información sobre cómo añadir dispositivos a zonas, consulte la Parte II, "Zonas" de *Guía de administración de sistemas: administración de recursos y contenedores de Oracle Solaris y zonas de Oracle Solaris*.

Uso de agrupaciones de almacenamiento de ZFS en una zona

Las agrupaciones de almacenamiento de ZFS no se pueden crear ni modificar en una zona. El modelo de administración delegada centraliza el control de dispositivos de almacenamiento físicos en la zona global y el control de almacenamiento virtual en zonas no globales. Aunque un conjunto de datos de agrupación se puede agregar a una zona, en una zona no se permite ningún comando que modifique las características físicas de la agrupación, por ejemplo crear, agregar o eliminar dispositivos. Aunque se agreguen dispositivos físicos a una zona mediante el comando `zonecfg` y el subcomando `add device`, o aunque se utilicen archivos, el comando `zpool` no permite la creación de agrupaciones en la zona.

Administración de propiedades de ZFS en una zona

Tras delegar un conjunto de datos a una zona, el administrador de zona puede controlar determinadas propiedades del conjunto. Cuando un conjunto de datos se delega a una zona, todos sus antecesores se ven como conjuntos de datos de sólo lectura, mientras que el conjunto de datos agregado y todos sus descendientes se pueden escribir. Por ejemplo, tenga en cuenta la configuración siguiente:

```
global# zfs list -Ho name
tank
tank/home
tank/data
tank/data/matrix
tank/data/zion
tank/data/zion/home
```

Si `tank/data/zion` se agrega a una zona, cada conjunto de datos tendrá las propiedades siguientes.

Conjunto de datos	Visible	Escribible	Propiedades invariables
`tank`	Sí	No	-
`tank/home`	No	-	-
`tank/data`	Sí	No	-
`tank/data/matrix`	No	-	-
`tank/data/zion`	Sí	Sí	`sharenfs`, `zoned`, `quota`, `reservation`
`tank/data/zion/home`	Sí	Sí	`sharenfs`, `zoned`

Cada conjunto primario de `tank/zone/zion` es visible como de sólo lectura, todos los descendientes se pueden escribir y los conjuntos de datos que no forman parte de la jerarquía superior no se ven. El administrador de zona no puede cambiar la propiedad `sharenfs` porque las zonas no globales no son válidas como servidores NFS. El administrador de zona tampoco puede cambiar la propiedad `zoned`; de lo contrario, habría un riesgo en la seguridad, como se explica en la sección siguiente.

Los usuarios con privilegios en la zona pueden cambiar otras propiedades configurables, excepto `quota` y `reservation`. Este comportamiento permite que el administrador de zona global controle la ocupación de espacio en el disco de todos los conjuntos de datos utilizados por la zona no global.

Asimismo, el administrador de zona global no puede modificar las propiedades `sharenfs` y `mountpoint` después de que un conjunto de datos se haya delegado a una zona no global.

Interpretación de la propiedad `zoned`

Si un conjunto de datos se delega a una zona no global, se debe marcar de modo especial para que determinadas propiedades no se interpreten en el contexto de la zona global. Tras haber delegado un conjunto de datos a una zona no global bajo el control de un administrador de zona, su contenido deja de ser fiable. Como en cualquier sistema de archivos, puede haber binarios `setuid`, vínculos simbólicos o contenido dudoso que podría repercutir negativamente en la seguridad de la zona global. Además, la propiedad `mountpoint` no se puede interpretar en el contexto de la zona global. Por otro lado, el administrador de zona podría afectar al espacio de nombre de la zona global. Para ocuparse de esto último, ZFS utiliza la propiedad `zoned` para indicar que un conjunto de datos se ha delegado a una zona no global en un determinado momento.

La propiedad `zoned` consiste en un valor booleano que se activa automáticamente la primera vez que se inicia una zona que contiene un conjunto de datos de ZFS. Un administrador de zona no tiene necesidad de activar manualmente esta propiedad. Si se establece la propiedad `zoned`, el conjunto de datos no se puede montar ni compartir en la zona global. En el ejemplo siguiente, `tank/zone/zion` se ha delegado a una zona y `tank/zone/global` no se ha delegado:

```
# zfs list -o name,zoned,mountpoint -r tank/zone
NAME                  ZONED  MOUNTPOINT
tank/zone/global        off  /tank/zone/global
tank/zone/zion           on  /tank/zone/zion
# zfs mount
tank/zone/global           /tank/zone/global
tank/zone/zion             /export/zone/zion/root/tank/zone/zion
```

Observe la diferencia entre la propiedad `mountpoint` y el directorio en que está montado el conjunto de datos `tank/zone/zion.` La propiedad `mountpoint` refleja la propiedad como almacenada en disco, no donde el conjunto de datos está montado en el sistema.

Si se elimina un conjunto de datos de una zona o se destruye una zona, la propiedad `zoned` *no* se elimina de forma automática. Este comportamiento se debe a los riegos de seguridad inherentes a estas tareas. Debido a que un usuario que no es de confianza dispone de acceso completo al conjunto de datos y sus descendientes, la propiedad `mountpoint` podría definirse con valores incorrectos o podría haber binarios `setuid` en los sistemas de archivos.

Para prevenir riesgos en la seguridad, el administrador de zona global debe suprimir manualmente la propiedad `zoned` si se desea volver a utilizar el conjunto de datos. Antes de establecer la propiedad `zoned` en `off`, compruebe que la propiedad `mountpoint` del conjunto de datos y todos sus descendientes tengan valores razonables y que no haya binarios `setuid`, o desactive la propiedad `setuid`.

Tras haber comprobado que no queden puntos débiles en la seguridad, la propiedad `zoned` se puede desactivar mediante los comandos `zfs set` o `zfs inherit`. Si la propiedad `zoned` se desactiva mientras un conjunto de datos se utiliza en una zona, el sistema podría manifestar un comportamiento impredecible. La propiedad se debe modificar únicamente si se tiene la certeza de que ninguna zona no global no está utilizando el conjunto de datos.

Uso de agrupaciones raíz de ZFS alternativas

Cuando se crea una agrupación, queda intrínsecamente vinculada con el sistema host. El sistema host mantiene información sobre la agrupación para detectar si está disponible o no. Si bien es útil en el funcionamiento normal, esta información puede suponer un obstáculo al iniciar desde otro soporte o al crear una agrupación en un soporte extraíble. Para solucionar este problema, ZFS proporciona una función de agrupaciones *raíz alternativas*. Una agrupación raíz alternativa no se mantiene de un reinicio del sistema a otro, y todos los puntos de montaje se modifican para vincularse con la raíz de la agrupación.

Creación de agrupaciones raíz de ZFS alternativas

La finalidad más habitual por la que se crea una agrupación raíz alternativa es utilizarla con soportes extraíbles. En estos casos, los usuarios suelen preferir un solo sistema de archivos, montado en algún lugar seleccionado en el sistema de destino. Si se crea una agrupación raíz alternativa mediante la opción `zpool create -R`, el punto de montaje del sistema de archivos raíz se establece automáticamente en `/`, que es el equivalente del valor raíz alternativo.

En el ejemplo siguiente, una agrupación denominada `morpheus` se crea con `/mnt` como ruta de acceso raíz alternativa:

```
# zpool create -R /mnt morpheus c0t0d0
# zfs list morpheus
NAME                  USED  AVAIL  REFER  MOUNTPOINT
morpheus             32.5K  33.5G     8K  /mnt
```

Observe el sistema de archivos único `morpheus`, cuyo punto de montaje es la raíz alternativa de la agrupación, `/mnt`. El punto de montaje que se guarda en disco es `/` y la ruta completa de `/mnt` sólo se interpreta en el contexto inicial de creación de la agrupación. Este sistema de archivos se puede exportar e importar bajo una agrupación raíz alternativa arbitraria en otro sistema, mediante sintaxis de *valor raíz alternativo* `-R`.

```
# zpool export morpheus
# zpool import morpheus
cannot mount '/': directory is not empty
# zpool export morpheus
# zpool import -R /mnt morpheus
# zfs list morpheus
NAME                  USED  AVAIL  REFER  MOUNTPOINT
morpheus             32.5K  33.5G     8K  /mnt
```

Importación de agrupaciones raíz alternativas

Las agrupaciones también se pueden importar mediante una raíz alternativa. Esta función posibilita situaciones de recuperación en que los puntos de montaje no se deben interpretar en el contexto de la raíz actual, sino en determinados directorios temporales en los que se pueden efectuar reparaciones. Esta función también es apta para montar soportes extraíbles como se ha explicado anteriormente.

En el ejemplo siguiente, una agrupación denominada `morpheus` se importa con `/mnt` como ruta de acceso raíz alternativa: En este ejemplo se da por sentado que `morpheus` se ha exportado previamente.

```
# zpool import -R /a pool
# zpool list morpheus
NAME   SIZE   ALLOC  FREE    CAP  HEALTH  ALTROOT
pool  44.8G    78K  44.7G     0%  ONLINE  /a
# zfs list pool
```

```
NAME   USED  AVAIL  REFER  MOUNTPOINT
pool  73.5K  44.1G    21K  /a/pool
```

Perfiles de derechos de ZFS

Si desea efectuar tareas de administración de ZFS sin utilizar la cuenta de superusuario (root), puede asumir una función con cualquiera de los perfiles siguientes para llevar a cabo dichas tareas de administración:

- Administración de almacenamiento de ZFS: proporciona privilegios para crear, destruir y manipular dispositivos en una agrupación de almacenamiento de ZFS
- Administración de sistemas de archivos ZFS: proporciona privilegios para crear, destruir y modificar sistemas de archivos ZFS

Para obtener más información sobre la creación o asignación de funciones, consulte *System Administration Guide: Security Services*.

Además de utilizar funciones RBAC para administrar sistemas de archivos ZFS, también puede considerar la posibilidad de utilizar la administración delegada de ZFS para tareas de administración ZFS distribuidas. Para más información, consulte el Capítulo 9, “Administración delegada de ZFS”.

◆ ◆ ◆ CAPÍTULO 11

Recuperación de agrupaciones y solución de problemas de Oracle Solaris ZFS

En este capítulo se explica la forma de identificar y solucionar errores de ZFS. También se proporciona información para la prevención de errores.

Este capítulo se divide en las secciones siguientes:

Identificación de errores de ZFS

Como combinación de sistema de archivos y administrador de volúmenes, ZFS puede presentar una amplia modalidad de errores. Este capítulo comienza con una breve introducción de los diversos errores y posteriormente explica el modo de identificarlos en un sistema que está en funcionamiento. Al final del capítulo, se proporcionan instrucciones para solucionar los problemas. ZFS puede tener tres tipos básicos de errores:

En una misma agrupación se pueden dar los tres errores, con lo cual un procedimiento completo de reparación implica detectar y corregir un error, luego ocuparse del siguiente error y así sucesivamente.

Dispositivos que faltan en una agrupación de almacenamiento de ZFS

Si un dispositivo ha desaparecido totalmente del sistema, ZFS detecta que dicho dispositivo no se puede abrir y le asigna el estado REMOVED. Según el nivel de repetición de datos que tenga la agrupación, la desaparición no tiene porqué significar que toda la agrupación deje de estar disponible. Si se elimina un disco de un dispositivo RAID-Z o reflejado, la agrupación sigue estando disponible. Una agrupación podría tener el estado `FAULTED`; esto significa que no será posible acceder a sus datos hasta que no se vuelva a colocar el dispositivo, en las condiciones detalladas a continuación:

- Si se eliminan todos los componentes de un reflejo
- Si se elimina más de un dispositivo en un RAID-Z (`raidz1`)
- Si se elimina un dispositivo de nivel superior en una configuración de un solo disco

Dispositivos dañados de una agrupación de almacenamiento de ZFS

El término "dañado" se aplica a una amplia diversidad de errores. Entre otros, están los errores siguientes:

- Errores transitorios de E/S debido a discos o controladores incorrectos
- Datos en disco dañados por rayos cósmicos
- Errores de controladores debidos a datos que se transfieren o reciben de ubicaciones incorrectas
- Anulación involuntaria de partes del dispositivo físico por parte de un usuario

En determinados casos, estos errores son transitorios, por ejemplo errores aleatorios de E/S mientras el controlador tiene problemas. En otros, las consecuencias son permanentes, por ejemplo la corrupción del disco. Aun así, el hecho de que los daños sean permanentes no implica necesariamente que el error se repita más adelante. Por ejemplo, si un administrador sobrescribe involuntariamente parte de un disco, no ha habido ningún error de hardware y no hace falta reemplazar el dispositivo. No resulta nada fácil identificar con exactitud lo que ha sucedido en un dispositivo. Ello se aborda en mayor profundidad más adelante en otra sección.

Datos dañados de ZFS

El deterioro de datos tiene lugar cuando uno o varios errores de dispositivos (dañados o que faltan) afectan a un dispositivo virtual de nivel superior. Por ejemplo, la mitad de un reflejo puede sufrir innumerables errores sin causar la más mínima corrupción de datos. Si se detecta un error en la misma ubicación de la otra parte del reflejo, habrá datos dañados.

Los datos quedan permanentemente dañados y deben tratarse de forma especial durante la reparación. Aunque se reparen o reemplacen los dispositivos subyacentes, los datos originales se pierden irremisiblemente. En estas circunstancias, casi siempre se requiere la restauración de datos a partir de copias de seguridad. Los errores de datos se registran conforme se detectan. Como se explica en la sección siguiente, pueden controlarse mediante limpiezas de agrupación rutinarias. Si se quita un bloque dañado, el siguiente pase de limpieza reconoce que el deterioro ya no está presente y suprime del sistema cualquier indicio de error.

Comprobación de integridad de sistema de archivos ZFS

En ZFS no hay una utilidad `fsck` equivalente. Esta utilidad se ha venido utilizando con dos fines: para reparaciones de sistema de archivos y para validaciones de dichos sistemas.

Reparación de sistema de archivos

En los sistemas de archivos tradicionales, el método de escritura de datos es intrínsecamente vulnerable a errores imprevistos que generan incoherencias en el sistema. Debido a que un sistema de archivos tradicional no es transaccional, puede haber bloques sin referenciar, recuentos de vínculos erróneos u otras estructuras de sistema de archivos no coherentes. La adición de diarios soluciona algunos de estos problemas, pero puede comportar otros si el registro no se puede invertir. La existencia de datos incoherentes en el disco de una configuración ZFS sólo puede ser debida a un error de hardware·(en cuyo caso, la agrupación debería haber sido redundante) o porque hay un error en el software de ZFS.

La utilidad `fsck` soluciona problemas conocidos específicos de sistemas de archivos UFS. Casi todos los problemas de agrupación de almacenamiento ZFS suelen estar relacionados con errores de hardware o fallos de alimentación. Muchos se pueden evitar utilizando agrupaciones redundantes. Si una agrupación se ha dañado por un error de hardware o un fallo de alimentación, consulte "Reparación de daños en las agrupaciones de almacenamiento de ZFS" en la página 317.

Si la agrupación no es redundante, siempre existe el riesgo de que los daños en el sistema de archivos lleguen a hacer que parte o todos los datos queden inaccesibles.

Validación de sistema de archivos

Aparte de reparar sistemas de archivos, la utilidad `fsck` comprueba que los datos en disco no tengan problemas. El procedimiento habitual para esta tarea consiste en desmontar el sistema de archivos y ejecutar la utilidad `fsck`, seguramente con el sistema en modo monousuario durante el proceso. Esta situación comporta un tiempo de inactividad proporcional al tamaño del sistema de archivos que se comprueba. En lugar de hacer que una determinada utilidad

realice la comprobación pertinente, ZFS brinda un mecanismo para ejecutar una comprobación rutinaria de todas las incoherencias. Esta función, denominada *limpieza*, se suele utilizar en la memoria y en otros sistemas como método para detectar y evitar errores antes de que deriven en errores de hardware o software.

Control de la limpieza de datos de ZFS

Cuando ZFS detecta un error, ya sea mediante el proceso de limpieza o al acceder a un archivo por algún motivo, el error se registra internamente para poder disponer de una visión general inmediata de todos los errores conocidos de la agrupación.

Limpieza explícita de datos de ZFS

La forma más sencilla de comprobar la integridad de los datos es ejecutar una limpieza explícita de todos los datos de la agrupación. Este proceso afecta a todos los datos de la agrupación y verifica que se puedan leer todos los bloques. El proceso de limpieza transcurre todo lo deprisa que permiten los dispositivos, aunque la prioridad de cualquier E/S quede por debajo de las operaciones normales. Esta operación puede incidir negativamente en el rendimiento, aunque los datos de la agrupación deberían seguir siendo utilizables casi del modo habitual. Para iniciar una limpieza explícita, utilice el comando `zpool scrub`. Por ejemplo:

```
# zpool scrub tank
```

El estado de la limpieza actual puede verse mediante el comando `zpool status`. Por ejemplo:

```
# zpool status -v tank
  pool: tank
 state: ONLINE
 scrub: scrub completed after 0h7m with 0 errors on Tue Tue Feb  2 12:54:00 2010
config:
        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0

errors: No known data errors
```

Sólo puede haber una operación de limpieza activa por agrupación.

Con la opción `-s` se puede detener una operación de limpieza en curso. Por ejemplo:

```
# zpool scrub -s tank
```

En la mayoría de los casos, una operación de limpieza para asegurar la integridad de los datos continúa hasta finalizar. Si cree que la limpieza afecta negativamente al rendimiento del sistema, puede detenerla.

La ejecución rutinaria de limpiezas garantiza la E/S continua en todos los discos del sistema. La ejecución rutinaria de limpiezas tiene el inconveniente de impedir que los discos inactivos pasen a la modalidad de bajo consumo. Si en general el sistema efectúa E/S permanentemente, o si el consumo de energía no es ningún problema, se puede prescindir de este tema.

Para obtener más información sobre la interpretación de la salida de `zpool status`, consulte "Consulta del estado de una agrupación de almacenamiento de ZFS" en la página 107.

Limpieza y actualización de la duplicación de datos de ZFS

Al reemplazar un dispositivo, se inicia una operación de actualización de duplicación de datos para transferir datos de las copias correctas al nuevo dispositivo. Este proceso es una forma de limpieza de disco. Por lo tanto, una acción de este tipo sólo puede darse en la agrupación en un momento determinado. Si hay una operación de limpieza en curso, una operación de actualización de duplicación de datos suspende la limpieza en curso y la reinicia una vez concluida la actualización de duplicación.

Para obtener más información sobre la actualización de duplicación de datos, consulte "Visualización del estado de la actualización de duplicación de datos" en la página 313.

Solución de problemas con ZFS

En las secciones siguientes se explica la manera de identificar y resolver problemas en los sistemas de archivos o agrupaciones de almacenamiento de ZFS:

Las funciones siguientes son válidas para identificar problemas en la configuración de ZFS:

- Se puede mostrar información detallada de agrupaciones de almacenamiento de ZFS utilizando el comando `zpool status`.
- Las notificaciones de errores en agrupaciones y dispositivos se realizan través de mensajes de diagnóstico de ZFS/FMA.
- Los comandos anteriores de ZFS que modificaban la información sobre el estado de las agrupaciones se ven ahora mediante el comando `zpool history`.

Casi todas las resoluciones de problemas de ZFS implican el uso del comando `zpool status`. Este comando analiza los errores de un sistema e identifica el problema más grave, sugiere una acción y proporciona un vínculo a documentación técnica para obtener más información. Aunque pueda haber varios problemas, el comando sólo identifica un problema de la agrupación. Por ejemplo, los errores de datos dañados generalmente denotan que ha fallado alguno de los dispositivos, pero la sustitución del dispositivo defectuoso podría no solucionar todos los problemas de deterioro de datos.

Además, un motor de diagnóstico de ZFS detecta y notifica errores de agrupaciones y dispositivos. También se notifican errores de suma de comprobación, E/S, dispositivos y agrupaciones asociados con errores de dispositivos o agrupaciones. Los errores de ZFS indicados por `fmd` se muestran en la consola y el archivo de mensajes del sistema. En la mayoría de los casos, el mensaje de `fmd` remite al comando `zpool status` para obtener más instrucciones sobre recuperación.

A continuación se expone el proceso básico de recuperación:

- Si procede, utilice el comando `zpool history` para identificar los comandos de ZFS anteriores que han desembocado en la situación de error. Por ejemplo:

  ```
  # zpool history tank
  History for 'tank':
  2010-07-15.12:06:50 zpool create tank mirror c0t1d0 c0t2d0 c0t3d0
  2010-07-15.12:06:58 zfs create tank/erick
  2010-07-15.12:07:01 zfs set checksum=off tank/erick
  ```

  Las sumas de comprobación de esta salida están inhabilitadas para el sistema de archivos `tank/erick`. No se recomienda esta configuración.
- Identifique los errores mediante los mensajes de `fmd` que aparecen en la consola del sistema o en el archivo `/var/adm/messages`.
- El comando `zpool status -x` proporciona más instrucciones de reparación.
- Repare los fallos, mediante las siguientes operaciones:
  - Sustitución del dispositivo defectuoso o ausente y conexión del mismo.
  - Restauración de la configuración defectuosa o los datos dañados a partir de una copia de seguridad.
  - Verificación de la recuperación mediante el comando `zpool status - x`.
  - Copia de seguridad de la configuración que se ha restaurado, si procede.

En esta sección se explica la forma de interpretar la salida `zpool status` para diagnosticar el tipo de fallos que se pueden producir. Si bien el comando ejecuta automáticamente casi todo el proceso, es importante comprender con exactitud los problemas que se identifican para poder diagnosticar el tipo de error. Las siguientes secciones describen cómo solucionar los diversos problemas que pueden producirse.

Cómo establecer si una agrupación de almacenamiento de ZFS tiene problemas

La forma más fácil de determinar si un sistema tiene problemas conocidos es mediante el comando `zpool status -x`. Este comando sólo describe agrupaciones que presentan problemas. Si no hay agrupaciones cuyo estado es defectuoso, el comando muestra lo siguiente:

```
# zpool status -x
all pools are healthy
```

Sin el indicador `-x`, el comando muestra el estado completo de todas las agrupaciones (o de la agrupación solicitada, si se indica en la línea de comandos), incluso si las agrupaciones están en buen estado.

Para obtener más información sobre las opciones de línea de comandos en la salida de `zpool status`, consulte "Consulta del estado de una agrupación de almacenamiento de ZFS" en la página 107.

Revisión de la salida de `zpool status`

La salida completa de `zpool status` se parece a la siguiente:

```
# zpool status tank
# zpool status tank
  pool: tank
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t1d0  UNAVAIL      0     0     0  cannot open

errors: No known data errors
```

Esta salida se describe a continuación:

Información sobre el estado general de la agrupación

Esta sección de la salida de `zpool status` contiene los campos siguientes (algunos de ellos sólo se muestran cuando hay agrupaciones con problemas):

`pool` El nombre de la agrupación.

`state` Estado actual de la agrupación. Esta información se refiere únicamente a la capacidad de la agrupación de proporcionar el nivel pertinente de repetición.

`status` Describe cuál es el problema que afecta a la agrupación. Si no se detectan errores, este campo se omite.

`action` Acción recomendada para la reparación de errores. Si no se detectan errores, este campo se omite.

`see`	Referencia a información técnica que contiene datos detallados sobre reparaciones. Los artículos en línea se actualizan con más frecuencia que esta guía. Por lo tanto, debe consultarlos para informarse sobre los procedimientos de reparación más recientes. Si no se detectan errores, este campo se omite.
`scrub`	Identifica el estado actual de una operación de limpieza, que puede contener la fecha y hora de conclusión de la última operación de limpieza, una limpieza en curso o si no se ha solicitado ninguna operación de limpieza.
`errors`	Identifica errores conocidos de datos o la ausencia de esta clase de errores.

Información de configuración de la agrupación

El campo `config` de la salida de `zpool status` describe la configuración de los dispositivos que conforman la agrupación, además de su estado y los posibles errores generados por los dispositivos. Puede mostrar uno de los estados siguientes: `ONLINE`, `FAULTED`, `DEGRADED`, `UNAVAIL` u `OFFLINE`. Si el estado es cualquiera de ellos menos `ONLINE`, significa que se pone el peligro la tolerancia a errores de la agrupación.

La segunda sección de la salida de configuración muestra estadísticas de errores. Dichos errores se dividen en tres categorías:

- `READ`: errores de E/S al emitir una solicitud de lectura
- `WRITE`: errores de E/S al emitir una solicitud de escritura
- `CKSUM`: errores de suma de comprobación, lo que significa que el dispositivo ha devuelto datos dañados como resultado de una solicitud de lectura

Estos errores son aptos para determinar si los daños son permanentes. Una cantidad pequeña de errores de E/S puede denotar un corte temporal del suministro; una cantidad grande puede denotar un problema permanente en el dispositivo. Estos errores no necesariamente corresponden a datos dañados según la interpretación de las aplicaciones. Si el dispositivo se encuentra en una configuración redundante, los dispositivos podrían mostrar errores irreparables, aunque no aparezcan errores en el reflejo o el nivel de dispositivos RAID-Z. En estos casos, ZFS ha recuperado correctamente los datos en buen estado e intentado reparar los datos dañados a partir de réplicas existentes.

Para obtener más información sobre la interpretación de estos errores, consulte "Cómo determinar el tipo de error en dispositivos" en la página 305.

En la última columna de la salida de `zpool status` se muestra información complementaria adicional. Dicha información se expande en el campo `state` para ayudar en el diagnóstico de modos de errores. Si un dispositivo tiene el estado `FAULTED`, este campo informa de si el dispositivo no está accesible o si dicho dispositivo tiene los datos dañados. Si se ejecuta la actualización de la duplicación de datos, el dispositivo muestra el progreso del proceso.

Para obtener información sobre el control del progreso de la actualización de duplicación de datos, consulte "Visualización del estado de la actualización de duplicación de datos" en la página 313.

Estado del proceso de limpieza

La sección de limpieza de la salida de `zpool status` describe el estado actual de cualquier operación de limpieza explícita. Esta información es diferente de si se detectan errores en el sistema, aunque es válida para determinar la exactitud de la información sobre datos dañados. Si la última operación de limpieza ha concluido correctamente, lo más probable es que se haya detectado cualquier tipo de datos dañados.

Los mensajes de limpieza completada se mantienen entre reinicios de sistema.

Para obtener más información sobre la limpieza de datos y la forma de interpretar esa información, consulte "Comprobación de integridad de sistema de archivos ZFS" en la página 295.

Errores de datos dañados

El comando `zpool status` muestra también si hay errores conocidos asociados con la agrupación. Estos errores se pueden haber detectado durante la limpieza de datos o en el transcurso del funcionamiento normal. ZFS mantiene un registro constante de todos los errores de datos asociados con una agrupación. El registro se reinicia cada vez que concluye una limpieza total del sistema.

Los errores de datos dañados siempre son fatales. El hecho de que existan denota que al menos una aplicación ha tenido un error de E/S debido a los datos dañados de la agrupación. Los errores de dispositivos en una agrupación redundante no generan datos dañados ni forman parte de este registro. De forma predeterminada, sólo se muestra el número de errores detectados. La opción `zpool status -v` proporciona una lista completa de errores con los detalles. Por ejemplo:

```
# zpool status -v
  pool: tank
 state: UNAVAIL
status: One or more devices are faulted in response to IO failures.
action: Make sure the affected devices are connected, then run 'zpool clear'.
   see: http://www.sun.com/msg/ZFS-8000-HC
 scrub: scrub completed after 0h0m with 0 errors on Tue Feb  2 13:08:42 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tank        UNAVAIL      0     0     0  insufficient replicas
          c1t0d0    ONLINE       0     0     0
          c1t1d0    UNAVAIL      4     1     0  cannot open

errors: Permanent errors have been detected in the following files:
```

```
/tank/data/aaa
/tank/data/bbb
/tank/data/ccc
```

El comando `fmd` muestra un mensaje parecido en la consola del sistema y el archivo `/var/adm/messages`. Con el comando `fmdump` se puede hacer un seguimiento de estos mensajes.

Para obtener más información sobre la interpretación de errores sobre corrupción de datos, consulte "Identificación del tipo de deterioro de datos" en la página 315.

Creación de informes del sistema sobre mensajes de error de ZFS

Aparte de hacer un constante seguimiento de los errores en la agrupación, ZFS muestra mensajes de `syslog` cuando se generan eventos de interés. Las situaciones siguientes generan eventos que notificar al administrador:

- **Transición de estados del dispositivo**: si un dispositivo pasa a tener el estado `FAULTED`, ZFS registra un mensaje que indica que la tolerancia a errores de la agrupación puede estar en peligro. Se envía un mensaje parecido si el dispositivo se conecta posteriormente, con lo cual la agrupación se recupera del error.
- **Datos dañados**: si se detecta cualquier tipo de datos dañados, ZFS registra un mensaje en el que se indica su ubicación y el momento en que tiene lugar. Este mensaje se registra sólo la primera vez que se detecta. Los accesos posteriores no generan ningún mensaje.
- **Errores de agrupaciones y de dispositivos**: si tiene lugar un error de agrupación o dispositivo, el daemon del administrador de errores informa de dichos errores mediante mensajes de `syslog` y mediante el comando `fmdump`.

Si ZFS detecta un error de dispositivo y se recupera automáticamente, no se genera ninguna notificación. Esta clase de errores no supone ningún fallo en la redundancia de la agrupación ni la integridad de los datos. Además, esta clase de errores suele ser fruto de un problema de controlador provisto de su propio conjunto de mensajes de error.

Reparación de una configuración de ZFS dañada

ZFS mantiene una caché de agrupaciones activas y su configuración en el sistema de archivos raíz. Si este archivo se daña o se desincroniza respecto a la información de configuración que se almacena en disco, no se podrá abrir la agrupación. ZFS procura evitar esta situación, si bien siempre se pueden producir daños arbitrarios debido a la naturaleza del almacenamiento subyacente. Al final termina desapareciendo una agrupación del sistema cuando lo normal es

que estuviera disponible. Esta situación también puede presentarse como una configuración parcial en la que falta un número no determinado de dispositivos virtuales de nivel superior. Sea como sea, la configuración se puede recuperar exportando la agrupación (si está visible) y volviéndola a importar.

Para obtener información sobre importación y exportación de agrupaciones, consulte "Migración de agrupaciones de almacenamiento de ZFS" en la página 116.

Resolución de un dispositivo que no se encuentra

Si no se puede abrir un dispositivo, se muestra como `UNAVAIL` en la salida de `zpool status`. Este estado indica que ZFS no ha podido abrir el dispositivo la primera vez que se accedió a la agrupación, o que desde entonces el dispositivo ya no está disponible. Si el dispositivo hace que no quede disponible un dispositivo virtual de alto nivel, la agrupación queda completamente inaccesible. De lo contrario, podría verse en peligro la tolerancia a errores de la agrupación. En cualquier caso, sólo tiene que volver a conectar el dispositivo al sistema para restablecer el funcionamiento normal.

Por ejemplo, en pantalla puede aparecer un mensaje parecido al siguiente procedente de `fmd` tras un error de dispositivo:

```
SUNW-MSG-ID: ZFS-8000-FD, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Thu Jun 24 10:42:36 PDT 2010
PLATFORM: SUNW,Sun-Fire-T200, CSN: -, HOSTNAME: neo2
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: a1fb66d0-cc51-cd14-a835-961c15696fcb
DESC: The number of I/O errors associated with a ZFS device exceeded
acceptable levels.  Refer to http://sun.com/msg/ZFS-8000-FD for more information.
AUTO-RESPONSE: The device has been offlined and marked as faulted.  An attempt
will be made to activate a hot spare if available.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.
```

Para ver información más pormenorizada del problema y la resolución, utilice el comando `zpool status -x`. Por ejemplo:

```
# zpool status -x
  pool: tank
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: scrub completed after 0h0m with 0 errors on Tue Feb  2 13:15:20 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
```

```
            c1t0d0  ONLINE       0     0     0
            c1t1d0  UNAVAIL      0     0     0  cannot open

errors: No known data errors
```

En esta salida puede observarse que el dispositivo `c1t1d0` ausente no funciona. Si determina que se trata de un dispositivo defectuoso, sustitúyalo.

A continuación, utilice el comando `zpool online` para conectar el dispositivo reemplazado. Por ejemplo:

```
# zpool online tank c1t1d0
```

Como último paso, confirme que la agrupación con el dispositivo reemplazado está en buen estado. Por ejemplo:

```
# zpool status -x tank
pool 'tank' is healthy
```

Cómo volver a conectar físicamente un dispositivo

La forma de volver a conectar un dispositivo que falta depende del tipo de dispositivo. Si es una unidad de red, se debe restaurar la conectividad a la red. Si se trata de un dispositivo USB u otro medio extraíble, debe volverse a conectar al sistema. Si consiste en un disco local, podría haber fallado un controlador de tal forma que el dispositivo ya no estuviera visible en el sistema. En tal caso, el controlador se debe reemplazar en el punto en que los discos vuelvan a estar disponibles. Pueden darse otros problemas, según el tipo de hardware y su configuración. Si una unidad falla y ya no está visible en el sistema, el dispositivo debe tratarse como si estuviera dañado. Siga los procedimientos que se indican en "Sustitución o reparación de un dispositivo dañado" en la página 305.

Notificación de ZFS sobre disponibilidad de dispositivos

Después de que un dispositivo se vuelve a conectar al sistema, ZFS puede detectar o no automáticamente su disponibilidad. Si la agrupación ya tenía errores o el sistema se reinició como parte del procedimiento de `conexión`, ZFS vuelve a explorar automáticamente todos los dispositivos cuando intenta abrir la agrupación. Si la agrupación se había degradado y el dispositivo se reemplazó cuando el sistema estaba en ejecución, se debe notificar a ZFS que el dispositivo ya está disponible y listo para abrirse de nuevo mediante el comando `zpool online`. Por ejemplo:

```
# zpool online tank c0t1d0
```

Para obtener más información sobre la conexión de dispositivos, consulte "Cómo conectar un dispositivo" en la página 95.

Sustitución o reparación de un dispositivo dañado

Esta sección describe la forma de determinar tipos de errores en dispositivos, eliminar errores transitorios y reemplazar un dispositivo.

Cómo determinar el tipo de error en dispositivos

El concepto *dispositivo dañado* es bastante ambiguo; puede referirse a varias situaciones:

- **Deterioro de bits**: con el tiempo, eventos aleatorios como campos magnéticos o rayos cósmicos pueden causar anomalías en los bits almacenados en el disco. Son eventos relativamente poco frecuentes, pero lo suficientemente habituales como para causar daños en datos de sistemas grandes o con procesos de larga duración.
- **Lecturas o escrituras de ubicaciones incorrectas**: los errores de firmware o hardware pueden hacer que lecturas o escrituras de bloques enteros hagan referencia a ubicaciones incorrectas en el disco. Suelen ser errores transitorios, pero si se producen en grandes cantidades podrían denotar una unidad defectuosa.
- **Error de administrador**: los administradores pueden sobrescribir inadvertidamente porciones del disco con datos dañados (por ejemplo, sobrescribir porciones de `/dev/zero` en el disco) que afectarán disco de manera permanente. Estos errores siempre son transitorios.
- **Interrupción temporal del suministro de energía**– Durante un determinado periodo de tiempo, quizá no se pueda acceder a un disco, lo que puede provocar errores de E/S. Esta situación se suele asociar con dispositivos conectados a redes, aunque los discos locales también pueden sufrir interrupciones temporales de suministro de energía. Estos errores pueden ser transitorios o no.
- **Hardware dañado o inestable**: esta situación constituye un cajón de sastre de todos los problemas que puede presentar un hardware defectuoso, entre los que se pueden citar errores persistentes de E/S y transportes defectuosos que causan errores aleatorios. Estos errores suelen ser permanentes.
- **Dispositivo sin conexión**: si un dispositivo está sin conexión, se supone que el administrador le ha asignado este estado porque presenta algún problema. El administrador que asigna este estado al dispositivo puede establecer si dicha suposición es correcta.

El diagnóstico exacto de la naturaleza del problema puede resultar un proceso complicado. El primer paso es examinar la cantidad de errores en la salida de `zpool status`. Por ejemplo:

```
# zpool status -v tpool
  pool: tpool
 state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption.  Applications may be affected.
action: Restore the file in question if possible.  Otherwise restore the
        entire pool from backup.
   see: http://www.sun.com/msg/ZFS-8000-8A
 scrub: scrub completed after 0h0m with 2 errors on Tue Jul 13 11:08:37 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tpool       ONLINE       2     0     0
          c1t1d0    ONLINE       2     0     0
          c1t3d0    ONLINE       0     0     0
errors: Permanent errors have been detected in the following files:

        /tpool/words
```

Los errores pueden ser de E/S o de suma de comprobación, y pueden denotar el posible tipo de defecto. El funcionamiento normal prevé muy pocos errores (sólo unos pocos en periodos de tiempo prolongados). Si detecta una gran cantidad de errores, probablemente denote la inminencia de un error o la inutilización completa de un dispositivo. Pero un error de administrador también puede derivar en grandes cantidades de errores. El registro del sistema `syslog` es la otra fuente de información. Si el registro tiene una gran cantidad de mensajes de controlador de canal de fibra o SCSI, es probable que la situación sea sintomática de graves problemas de hardware. Si no se generan mensajes de `syslog`, es probable que los daños sean transitorios.

El objetivo es responder a la pregunta siguiente:

¿Es probable que este dispositivo vuelva a tener un error?

Los errores que suceden sólo una vez se consideran *transitorios* y no denotan problemas potenciales. Los errores continuos o suficientemente graves como para indicar problemas potenciales en el hardware se consideran errores *fatales*. El hecho de determinar el tipo de error trasciende el ámbito de cualquier software automatizado que haya actualmente en ZFS, por lo cual eso es una tarea propia de los administradores. Una vez determinado el error, se puede llevar a cabo la acción pertinente. Suprima los errores transitorios o reemplace los dispositivos con errores fatales. Estos procedimientos de reparación se explican en las secciones siguientes.

Aun en caso de que los errores de dispositivos se consideren transitorios, se pueden haber generado errores incorregibles en los datos de la agrupación. Estos errores precisan procedimientos especiales de reparación, incluso si el dispositivo subyacente se considera que está en buen estado o se ha reparado. Para obtener más información sobre cómo reparar errores de datos, consulte "Reparación de datos dañados" en la página 314.

Supresión de errores transitorios

Si los errores en dispositivos se consideran transitorios, en el sentido de que es poco probable que incidan más adelante en el buen estado del dispositivo, se pueden suprimir tranquilamente para indicar que no se ha producido ningún error fatal. Para suprimir los recuentos de errores de RAID-Z o dispositivos reflejados, utilice el comando `zpool clear`. Por ejemplo:

```
# zpool clear tank c1t1d0
```

Esta sintaxis suprime todos los errores de dispositivo y recuentos de errores de datos asociados con el dispositivo.

Utilice la sintaxis siguiente para suprimir todos los errores asociados con los dispositivos virtuales de una agrupación y para suprimir los recuentos de errores de datos asociados con la agrupación:

```
# zpool clear tank
```

Para obtener más información sobre la supresión de errores de dispositivos, consulte "Borrado de errores de dispositivo de agrupación de almacenamiento" en la página 96.

Sustitución de un dispositivo de una agrupación de almacenamiento de ZFS

Si los daños en un dispositivo son permanentes o es posible que lo sean en el futuro, dicho dispositivo debe reemplazarse. El hecho de que el dispositivo pueda sustituirse o no depende de la configuración.

Cómo determinar si un dispositivo se puede reemplazar o no

Para que un dispositivo se pueda reemplazar, la agrupación debe tener el estado `ONLINE`. El dispositivo debe formar parte de una configuración redundante o tener el estado correcto (en estado `ONLINE`). Si el dispositivo forma parte de una configuración redundante, debe haber suficientes réplicas desde las que poder recuperar datos en buen estado. Si dos discos con reflejo de cuatro vías son defectuosos, se puede reemplazar cualquiera de ellos porque se dispone de réplicas en buen estado. Sin embargo, si hay dos discos defectuosos en un dispositivo virtual RAID-Z (`raidz1`) de cuatro vías, ninguno de ellos se puede reemplazar porque no se dispone de suficientes réplicas desde las que recuperar datos. Si el dispositivo está dañado pero tiene

conexión, se puede reemplazar siempre y cuando la agrupación no tenga el estado FAULTED. Sin embargo, cualquier dato dañado del dispositivo se copia al nuevo dispositivo a menos que haya suficientes réplicas con datos correctos.

En la configuración siguiente, el disco c1t1d0 se puede reemplazar y los datos de la agrupación se copian de la réplica en buen estado, c1t0d0.

```
mirror            DEGRADED
c1t0d0             ONLINE
c1t1d0             FAULTED
```

El disco c1t0d0 también se puede reemplazar, aunque no es factible la recuperación automática de datos debido a la falta de réplicas en buen estado.

En la configuración siguiente, no se puede reemplazar ninguno de los discos dañados. Los discos con el estado ONLINE tampoco pueden reemplazarse porque la agrupación está dañada.

```
raidz              FAULTED
c1t0d0             ONLINE
c2t0d0             FAULTED
c3t0d0             FAULTED
c4t0d0             ONLINE
```

En la configuración siguiente, el disco de nivel superior tampoco se puede reemplazar, si bien en el disco nuevo se va a copiar cualquier dato dañado.

```
c1t0d0         ONLINE
c1t1d0         ONLINE
```

Si cualquiera de los discos es defectuoso, no se puede reemplazar porque la agrupación también está dañada.

Dispositivos que no se pueden reemplazar

Si la pérdida de un dispositivo causa el deterioro de la agrupación, o el dispositivo contiene demasiados errores en los datos en una configuración no redundante, la correcta sustitución del dispositivo no es factible. Si la redundancia es insuficiente, no es posible restaurar con datos en buen estado el dispositivo dañado. En este caso, la única posibilidad es destruir la agrupación, volver a crear la configuración y, a continuación, restaurar los datos desde una copia de seguridad.

Para obtener más información sobre cómo restaurar toda una agrupación, consulte "Reparación de daños en las agrupaciones de almacenamiento de ZFS" en la página 317.

Sustitución de un dispositivo de una agrupación de almacenamiento de ZFS

Tras determinar que se puede reemplazar un dispositivo, utilice el comando `zpool replace` para reemplazarlo. Si va a reemplazar el dispositivo dañado con otro diferente, utilice sintaxis como ésta:

```
# zpool replace tank c1t1d0 c2t0d0
```

Este comando migra datos al dispositivo nuevo desde el dispositivo dañado, o de otros dispositivos de la agrupación si la configuración es redundante. Cuando finaliza el comando, desconecta el dispositivo dañado de la configuración. Es entonces cuando el dispositivo se puede eliminar del sistema. Si ya ha eliminado el dispositivo y lo ha reemplazado por uno nuevo en la misma ubicación, utilice la forma de un solo dispositivo del comando. Por ejemplo:

```
# zpool replace tank c1t1d0
```

Este comando selecciona un disco sin formato, le aplica el formato correspondiente y actualiza la duplicación de datos a partir del resto de la configuración.

Para obtener más información acerca del comando `zpool replace`, consulte "Sustitución de dispositivos en una agrupación de almacenamiento" en la página 96.

EJEMPLO 11–1 Sustitución de un dispositivo de una agrupación de almacenamiento de ZFS

El ejemplo siguiente muestra cómo reemplazar un dispositivo (`c1t3d0`) en la agrupación de almacenamiento reflejada `tank` en un sistema Sun Fire x4500 de Oracle. Para reemplazar el disco `c1t3d0` con un nuevo disco en la misma ubicación (`c1t3d0`), desconfigure el disco antes de intentar reemplazarlo. Los pasos básicos son:

- Desconectar el disco (`c1t3d0`) que se va a sustituir. No puede desconfigurar un disco que se esté utilizando.
- Utilizar el comando `cfgadm` para identificar el disco (`c1t3d0`) que desconfigurar y desconfigurarlo. La agrupación se degradará con el disco desconectado en esta configuración reflejada, pero la agrupación seguirá estando disponible.
- Sustituir físicamente el disco (`c1t3d0`). Antes de quitar la unidad que falla, asegúrese de que se encienda el LED azul que indica que el disco está `listo para quitar`.
- Volver a configurar el disco (`c1t3d0`).
- Conectar el disco nuevo (`c1t3d0`).
- Ejecutar el comando `zpool replace` para reemplazar el disco (`c1t3d0`).

EJEMPLO 11–1 Sustitución de un dispositivo de una agrupación de almacenamiento de ZFS *(Continuación)*

Nota – Si ha configurado previamente la propiedad de agrupación `autoreplace` como `on`, se dará formato y se sustituirá automáticamente cualquier dispositivo nuevo que se detecte en la misma ubicación física como dispositivo que antes pertenecía a la agrupación, mediante el comando `zpool replace`. Es posible que el hardware no sea compatible con esta función.

- Si un disco fallido se sustituye automáticamente por un repuesto en marcha, puede que deba desconectarlo después de dicha sustitución. Por ejemplo, si `c2t4d0` es aún un repuesto en marcha activo después de sustituir el disco fallido, desconéctelo.

  ```
  # zpool detach tank c2t4d0
  ```

El ejemplo siguiente detalla los pasos para reemplazar un disco en una agrupación de almacenamiento de ZFS.

```
# zpool offline tank c1t3d0
# cfgadm | grep c1t3d0
sata1/3::dsk/c1t3d0            disk         connected    configured   ok
# cfgadm -c unconfigure sata1/3
Unconfigure the device at: /devices/pci@0,0/pci1022,7458@2/pci11ab,11ab@1:3
This operation will suspend activity on the SATA device
Continue (yes/no)? yes
# cfgadm | grep sata1/3
sata1/3                        disk         connected    unconfigured ok
<Physically replace the failed disk c1t3d0>
# cfgadm -c configure sata1/3
# cfgadm | grep sata1/3
sata1/3::dsk/c1t3d0            disk         connected    configured   ok
# zpool online tank c1t3d0
# zpool replace tank c1t3d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:17:32 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c1t1d0  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            c0t2d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
            c0t3d0  ONLINE       0     0     0
            c1t3d0  ONLINE       0     0     0

errors: No known data errors
```

EJEMPLO 11–1 Sustitución de un dispositivo de una agrupación de almacenamiento de ZFS *(Continuación)*

Tenga en cuenta que el comando `zpool output` anterior podría mostrar tanto los discos nuevos como los antiguos en un encabezado *replacing*. Por ejemplo:

```
replacing     DEGRADED     0     0    0
  c1t3d0s0/o  FAULTED      0     0    0
  c1t3d0      ONLINE       0     0    0
```

Este texto indica que el proceso de sustitución está en curso y se está actualizando la duplicación de datos.

Si va a reemplazar un disco (`c1t3d0`) con otro (`c4t3d0`), sólo tiene que ejecutar el comando `zpool replace`. Por ejemplo:

```
# zpool replace tank c1t3d0 c4t3d0
# zpool status
  pool: tank
 state: DEGRADED
 scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:

        NAME             STATE     READ WRITE CKSUM
        tank             DEGRADED     0     0     0
          mirror-0       ONLINE       0     0     0
            c0t1d0       ONLINE       0     0     0
            c1t1d0       ONLINE       0     0     0
          mirror-1       ONLINE       0     0     0
            c0t2d0       ONLINE       0     0     0
            c1t2d0       ONLINE       0     0     0
          mirror-2       DEGRADED     0     0     0
            c0t3d0       ONLINE       0     0     0
            replacing    DEGRADED     0     0     0
              c1t3d0     OFFLINE      0     0     0
              c4t3d0     ONLINE       0     0     0

errors: No known data errors
```

Es posible que deba ejecutar el comando `zpool status` varias veces hasta finalizar la sustitución del disco.

```
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:

        NAME          STATE     READ WRITE CKSUM
        tank          ONLINE       0     0     0
          mirror-0    ONLINE       0     0     0
            c0t1d0    ONLINE       0     0     0
            c1t1d0    ONLINE       0     0     0
          mirror-1    ONLINE       0     0     0
```

EJEMPLO 11–1 Sustitución de un dispositivo de una agrupación de almacenamiento de ZFS *(Continuación)*

```
            c0t2d0    ONLINE       0     0     0
            c1t2d0    ONLINE       0     0     0
          mirror-2    ONLINE       0     0     0
            c0t3d0    ONLINE       0     0     0
            c4t3d0    ONLINE       0     0     0
```

EJEMPLO 11–2 Sustitución de un dispositivo de registro que presenta errores

El ejemplo siguiente muestra cómo recuperar un dispositivo de registro (`c0t5d0`) que presenta errores en la agrupación de almacenamiento, (`pool`). Los pasos básicos son:

- Revisar el resultado de `zpool status -x` y el mensaje de diagnóstico de FMA que se describen a continuación:

  `http://www.sun.com/msg/ZFS-8000-K4`

- Reemplazar físicamente el dispositivo de registro que presenta errores.
- Conectar el dispositivo de registro.
- Borrar la condición de error de la agrupación.

```
# zpool status -x
  pool: pool
 state: FAULTED
status: One or more of the intent logs could not be read.
        Waiting for adminstrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
        or ignore the intent log records by running 'zpool clear'.
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        pool        FAULTED      0     0     0 bad intent log
          mirror    ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
        logs        FAULTED      0     0     0 bad intent log
          c0t5d0    UNAVAIL      0     0     0 cannot open
<Physically replace the failed log device>
# zpool online pool c0t5d0
# zpool clear pool

# zpool status -x
  pool: pool
 state: FAULTED
status: One or more of the intent logs could not be read.
        Waiting for adminstrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
        or ignore the intent log records by running 'zpool clear'.
 scrub: none requested
config:
```

EJEMPLO 11–2 Sustitución de un dispositivo de registro que presenta errores *(Continuación)*

```
        NAME          STATE     READ WRITE CKSUM
        pool          FAULTED      0     0     0 bad intent log
          mirror-0    ONLINE       0     0     0
            c0t1d0    ONLINE       0     0     0
            c0t4d0    ONLINE       0     0     0
        logs          FAULTED      0     0     0 bad intent log
          c0t5d0      UNAVAIL      0     0     0 cannot open
<Physically replace the failed log device>
# zpool online pool c0t5d0
# zpool clear pool
```

Visualización del estado de la actualización de duplicación de datos

El proceso de reemplazar un dispositivo puede tardar una considerable cantidad de tiempo, según el tamaño del dispositivo y la cantidad de datos que haya en la agrupación. El proceso de transferir datos de un dispositivo a otro, denominado *actualización de la duplicación de datos*, se puede controlar mediante el comando `zpool status`.

Los sistemas de archivos tradicionales actualizan duplicaciones de datos en los bloques. Debido a que ZFS suprime la disposición artificial de capas de Volume Manager, puede ejecutar la actualización de duplicación de datos de manera más potente y controlada. Esta función presenta dos ventajas principales:

- ZFS sólo actualiza la duplicación de los datos necesarios. En caso de una breve interrupción del suministro (en contraposición a un reemplazo completo del dispositivo), la actualización de duplicación de datos del disco puede hacerse en cuestión de segundos, Si se reemplaza todo un disco, el tiempo que implica el proceso de actualización de duplicación de datos es proporcional a la cantidad de datos que se utilizan en disco. La sustitución de un disco de 500 GB puede ser cuestión de segundos si la agrupación sólo tiene unos cuantos gigabytes de espacio usado en el disco.
- La actualización de duplicación de datos es un proceso seguro que se puede interrumpir. Si el sistema se queda sin conexión o se reinicia, el proceso de actualización de duplicación de datos reanuda la tarea exactamente en el punto en que se había interrumpido, sin que haga falta hacer nada.

Para observar el progreso de la actualización de duplicación de datos, utilice el comando `zpool status`. Por ejemplo:

```
# zpool status tank
  pool: tank
 state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
 scrub: resilver in progress for 0h0m, 22.60% done, 0h1m to go
config:
        NAME                  STATE     READ WRITE CKSUM
```

```
        tank             DEGRADED     0     0     0
          mirror-0       DEGRADED     0     0     0
            replacing-0  DEGRADED     0     0     0
              c1t0d0     UNAVAIL      0     0     0  cannot open
              c2t0d0     ONLINE       0     0     0  85.0M resilvered
            c1t1d0       ONLINE       0     0     0

errors: No known data errors
```

En este ejemplo, el disco `c1t0d0` se sustituye por `c2t0d0`. Este evento se refleja en la salida del estado mediante la presencia del dispositivo virtual que `reemplaza` en la configuración. Este dispositivo no es real ni sirve para crear una agrupación. La única finalidad de este dispositivo es mostrar el proceso de actualización de duplicación de datos e identificar el dispositivo que se va a reemplazar.

Cualquier agrupación sometida al proceso de actualización de duplicación de datos adquiere el estado `ONLINE` o `DEGRADED`, porque hasta que no haya finalizado dicho proceso es incapaz de proporcionar el nivel necesario de redundancia. La actualización de duplicación de datos se ejecuta lo más deprisa posible, si bien la E/S siempre se programa con una prioridad inferior a la E/S solicitada por el usuario, para que repercuta en el sistema lo menos posible. Tras finalizarse la actualización de duplicación de datos, la configuración asume los parámetros nuevos. Por ejemplo:

```
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: resilver completed after 0h1m with 0 errors on Tue Feb  2 13:54:30 2010
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0  377M resilvered
            c1t1d0  ONLINE       0     0     0

errors: No known data errors
```

La agrupación pasa de nuevo al estado `ONLINE` y el disco dañado original (`c1t0d0`) desaparece de la configuración.

Reparación de datos dañados

En las secciones siguientes se explica el procedimiento para identificar el tipo de corrupción de datos y, si es factible, cómo reparar los datos.

- "Identificación del tipo de deterioro de datos" en la página 315
- "Reparación de un archivo o directorio dañado" en la página 316
- "Reparación de daños en las agrupaciones de almacenamiento de ZFS" en la página 317

Para reducir al mínimo las posibilidades de que los datos sufran daños, ZFS utiliza sumas de comprobación, redundancia y datos que se reparan a sí mismos. Ahora bien, los datos se pueden dañar si una agrupación no es redundante, cuando una agrupación está en estado "degraded" o si se combina una improbable serie de eventos para dañar varias copias de determinados datos. Sea cual sea el origen, el resultado es el mismo: los datos quedan dañados y no se puede acceder a ellos. Las medidas requeridas dependen del tipo de datos dañados y su valor relativo. Se pueden dañar dos tipos básicos de datos:

- Metadatos de agrupación: para abrir una agrupación y acceder a conjuntos de datos, ZFS debe analizar cierta cantidad de datos. Si se dañan estos datos, quedará inaccesible toda la agrupación o partes de la jerarquía del conjuntos de datos.
- Datos de objeto: en este caso, el daño afecta a un determinado archivo o directorio. Ello puede hacer que no sea posible acceder a una parte del archivo o directorio, o causar la interrupción del objeto.

Los datos se verifican durante el funcionamiento normal y durante el proceso de limpieza. Para obtener más información sobre cómo verificar la integridad de datos de agrupaciones, consulte "Comprobación de integridad de sistema de archivos ZFS" en la página 295.

Identificación del tipo de deterioro de datos

De forma predeterminada, el comando `zpool status` avisa únicamente de la presencia de daños, pero no indica su ubicación. Por ejemplo:

```
# zpool status monkey
  pool: monkey
 state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption.  Applications may be affected.
action: Restore the file in question if possible.  Otherwise restore the
        entire pool from backup.
   see: http://www.sun.com/msg/ZFS-8000-8A
 scrub: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:

        NAME        STATE     READ WRITE CKSUM
        monkey      ONLINE       8     0     0
          c1t1d0    ONLINE       2     0     0
          c2t5d0    ONLINE       6     0     0

errors: 8 data errors, use '-v' for a list
```

Cada error indica solamente que ha habido un error en un determinado momento. Eso no significa que cada error siga estando en el sistema. Éste es el caso en circunstancias normales. Determinadas interrupciones temporales del suministro pueden comportar daños en los datos que se reparan automáticamente cuando finaliza dicha interrupción. Se garantiza la ejecución completa de un proceso de limpieza de la agrupación para examinar cada bloque activo de la

agrupación, con lo cual el registro de errores se reinicia cuando concluye la limpieza. Si considera que ya no hay errores y no quiere esperar a que finalice la limpieza, reinicie todos los errores de la agrupación mediante el comando `zpool online`.

Si los dañados afectan a metadatos de toda la agrupación, la salida difiere ligeramente. Por ejemplo:

```
# zpool status -v morpheus
  pool: morpheus
    id: 1422736890544688191
 state: FAULTED
status: The pool metadata is corrupted.
action: The pool cannot be imported due to damaged devices or data.
   see: http://www.sun.com/msg/ZFS-8000-72
config:

        morpheus    FAULTED   corrupted data
          c1t10d0   ONLINE
```

Si los daños afectan a toda la agrupación, ésta pasa al estado `FAULTED` , ya que posiblemente no podrá proporcionar el nivel de redundancia requerido.

Reparación de un archivo o directorio dañado

Si un archivo o directorio resultasen dañados, según el tipo de corrupción, el sistema podría seguir funcionando. Si en el sistema no hay copias de los datos de buena calidad, cualquier daño que tenga lugar será irreparable. Si los datos son importantes, la única alternativa es recuperarlos a partir de una copia de seguridad. Aun así, debe poder realizar una recuperación sin necesidad de restaurar toda la agrupación.

Si se ha dañado un bloque datos de archivo, el archivo se puede eliminar sin problemas; de este modo, el error desaparece del sistema. Utilice el comando `zpool status -v` para ver en pantalla una lista con nombres de archivos que tienen errores constantes. Por ejemplo:

```
# zpool status -v
  pool: monkey
 state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption.  Applications may be affected.
action: Restore the file in question if possible.  Otherwise restore the
        entire pool from backup.
   see: http://www.sun.com/msg/ZFS-8000-8A
 scrub: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:

        NAME        STATE     READ WRITE CKSUM
        monkey      ONLINE       8     0     0
          c1t1d0    ONLINE       2     0     0
          c2t5d0    ONLINE       6     0     0
```

```
errors: Permanent errors have been detected in the following files:

/monkey/a.txt
/monkey/bananas/b.txt
/monkey/sub/dir/d.txt
monkey/ghost/e.txt
/monkey/ghost/boo/f.txt
```

La lista de nombres de archivos con errores constantes se puede describir del modo siguiente:

- Si se busca la ruta de acceso del archivo y se monta el conjunto de datos, se muestra en pantalla toda la ruta del archivo. Por ejemplo:

  ```
  /monkey/a.txt
  ```

- Si se busca la ruta de acceso del archivo pero el conjunto de datos no se monta, en pantalla se muestra el nombre del conjunto de datos sin una barra inclinada (/), seguido de la ruta de acceso del conjunto de datos al archivo. Por ejemplo:

  ```
  monkey/ghost/e.txt
  ```

- Si no se puede trasladar correctamente el número de objeto a una ruta de archivo, ya sea por un error o porque el objeto no tiene asociada ninguna ruta de archivo auténtica, como en el caso de `dnode_t`, en pantalla se muestra nombre del conjunto de datos seguido del número de objeto. Por ejemplo:

  ```
  monkey/dnode:<0x0>
  ```

- Si se daña un objeto del conjunto de metaobjetos, en pantalla se muestra un etiqueta especial de `<metadata>`, seguida del número de objeto.

Si los daños se dan en un directorio o en los metadatos de un archivo, la única alternativa es colocar el archivo en otra ubicación. Puede colocar cualquier archivo o directorio en una ubicación menos apropiada para poder restaurar el objeto original.

Reparación de daños en las agrupaciones de almacenamiento de ZFS

Si los metadatos de una agrupación resultan dañados de forma que no es posible abrir la agrupación o importarla, puede optar por:

- Intentar recuperar la agrupación mediante uno de los comandos `zpool clear - F` o `zpool import -F`. Estos comandos intentan restaurar un estado operativo de las transacciones de agrupación más recientes. Puede utilizar el comando `zpool status` para revisar una agrupación dañada y el procedimiento de recuperación recomendado. Por ejemplo:

  ```
  # zpool status
    pool: tpool
   state: FAULTED
  status: The pool metadata is corrupted and the pool cannot be opened.
  action: Recovery is possible, but will result in some data loss.
  ```

```
        Returning the pool to its state as of Wed Jul 14 11:44:10 2010
        should correct the problem.  Approximately 5 seconds of data
        must be discarded, irreversibly.  Recovery can be attempted
        by executing 'zpool clear -F tpool'.  A scrub of the pool
        is strongly recommended after recovery.
   see: http://www.sun.com/msg/ZFS-8000-72
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
         tpool      FAULTED      0     0     1  corrupted data
          c1t1d0    ONLINE       0     0     2
          c1t3d0    ONLINE       0     0     4
```

El proceso de recuperación descrito anteriormente consiste en utilizar el comando:

```
# zpool clear -F tpool
```

Si intenta importar una agrupación de almacenamiento dañada, se muestran mensajes parecidos al siguiente:

```
# zpool import tpool
cannot import 'tpool': I/O error
        Recovery is possible, but will result in some data loss.
        Returning the pool to its state as of Wed Jul 14 11:44:10 2010
        should correct the problem.  Approximately 5 seconds of data
        must be discarded, irreversibly.  Recovery can be attempted
        by executing 'zpool import -F tpool'.  A scrub of the pool
        is strongly recommended after recovery.
```

El proceso de recuperación descrito anteriormente consiste en utilizar el comando:

```
# zpool import -F tpool
Pool tpool returned to its state as of Wed Jul 14 11:44:10 2010.
Discarded approximately 5 seconds of transactions
```

Si la agrupación dañada está en el archivo `zpool.cache`, el problema se descubre al iniciar el sistema, y dicha agrupación se notifica en el comando `zpool status`. Si la agrupación no está en el archivo `zpool.cache`, no se importará o abrirá correctamente y aparecerán mensajes de agrupación dañada al intentar importarla.

- Si la agrupación no se puede recuperar con el método de recuperación descrito anteriormente, deberá restaurar la agrupación y todos sus datos desde una copia de seguridad. Los procedimientos para ello son muy variados: dependen de la configuración de las agrupaciones y de la estrategia de las copias de seguridad. En primer lugar, guarde la configuración tal como aparece en el comando `zpool status` para poder crearla de nuevo después de destruida la agrupación. A continuación, utilice el comando `zpool destroy -f` para destruir la agrupación. Asimismo, conserve un archivo que contenga la disposición de los conjuntos de datos y guarde en lugar seguro las distintas propiedades que se han definido, ya que si en algún momento no se puede acceder a la agrupación, tampoco se podrá acceder a esta información. A partir de la configuración de la agrupación y la

disposición del conjunto de datos, es posible reconstruir toda la configuración tras la destrucción de la agrupación. Los datos se pueden rellenar utilizando cualquier método de restauración o copia de seguridad.

Reparación de un sistema que no se puede iniciar

ZFS se ha concebido para mantenerse robusto y estable frente a los errores. Aun así, los errores de software o problemas imprevistos pueden desequilibrar el sistema al intentar acceder a una agrupación. Como parte del proceso de inicio se debe abrir cada agrupación, lo que significa que esta clase de errores harán que el sistema entre en un bucle de inicios de emergencia. Para solucionar esta situación, debe indicar a ZFS que no busque agrupaciones al inicio.

ZFS mantiene una caché interna de agrupaciones disponibles junto con sus configuraciones en `/etc/zfs/zpool.cache`. La ubicación y el contenido de este archivo son personales y susceptibles de cambiarse. Si el sistema no se puede iniciar, inicie en `none` mediante la opción `-m =none`. Cuando el sistema se haya iniciado, vuelva a montar el sistema de archivos raíz como grabable y cambie el nombre o cambie la ubicación del archivo `/etc/zfs/zpool.cache`. Estas acciones hacen que ZFS no tenga en cuenta que en el sistema hay agrupaciones, lo cual impide que intente acceder a la agrupación dañada que causa el problema. A continuación, puede pasar a un estado normal del sistema mediante el comando `svcadm milestone all`. Se puede aplicar un proceso similar al iniciar desde un sistema de archivos raíz alternativo para efectuar reparaciones.

Cuando el sistema se haya iniciado, puede intentar importar la agrupación mediante el comando `zpool import`. Sin embargo, es probable que se cause el mismo error que al inicio, puesto que el comando emplea el mismo mecanismo de acceso a agrupaciones. Si en el sistema hay varias agrupaciones, haga lo siguiente:

- Cambie el nombre de `zpool.cache` o lleve el archivo a otra ubicación, tal como se ha indicado anteriormente.
- Determine qué agrupación podría tener problemas utilizando el comando `fmdump -eV`, para ver las agrupaciones que han notificado errores fatales.
- Importe una por una las agrupaciones que tienen problemas, como se describe en la salida de `fmdump`.

◆ ◆ ◆ APÉNDICE A

Descripciones de versiones de Oracle Solaris ZFS

Este apéndice describe versiones de ZFS disponibles, las características de cada versión y el sistema operativo Solaris pertinente.

Este apéndice contiene las secciones siguientes:

Información general de versiones de ZFS

El uso de una versión concreta de ZFS disponible en versiones de Solaris da acceso a nuevas funciones de sistema de archivos y agrupación de ZFS. Puede utilizar uno de los comandos `zpool upgrade` o `zfs upgrade` para identificar si una agrupación o un sistema de archivos es de una versión anterior a la suministrada con la versión de Solaris vigente. También puede usar estos comandos para actualizar sus versiones de sistema de archivos y agrupación.

Para obtener información sobre el uso de los comandos `zpool upgrade` y `zfs upgrade`, consulte "Actualización de sistemas de archivos ZFS (`zfs upgrade`)" en la página 33 y "Actualización de agrupaciones de almacenamiento de ZFS" en la página 123.

Versiones de agrupación de ZFS

La siguiente tabla proporciona una lista de versiones de agrupación de ZFS disponibles en las versiones de Solaris.

Versión	Solaris 10	Descripción
1	Solaris 10 6/06	Versión ZFS inicial

Versión	Solaris 10	Descripción
2	Solaris 10 11/06	Bloques ditto (metadatos repetidos)
3	Solaris 10 11/06	Repuestos en marcha y RAID-Z de doble paridad
4	Solaris 10 8/07	`zpool history`
5	Solaris 10 10/08	Algoritmo de compresión `gzip`
6	Solaris 10 10/08	propiedad de agrupación `bootfs`
7	Solaris 10 10/08	Dispositivos de registro con diferente función
8	Solaris 10 10/08	Administración delegada
9	Solaris 10 10/08	propiedades `refquota` y `refreservation`
10	Solaris 10 5/09	Dispositivos de caché
11	Solaris 10 10/09	Rendimiento de limpieza mejorado
12	Solaris 10 10/09	Propiedades de instantáneas
13	Solaris 10 10/09	Propiedad `Snapused`
14	Solaris 10 10/09	Propiedad `aclinherit passthrough-x`
15	Solaris 10 10/09	cálculo de espacio de agrupación y usuario
16	Solaris 10 9/10	compatibilidad de propiedad `stmf`
17	Solaris 10 9/10	RAID-Z de triple paridad
18	Solaris 10 9/10	Retenciones de instantánea
19	Solaris 10 9/10	Eliminación de dispositivo de registro
20	Solaris 10 9/10	`Compression using zle` (codificación de caracteres de longitud cero)
21	Solaris 10 9/10	Reservado
22	Solaris 10 9/10	Propiedades recibidas

Versiones de sistema de archivos ZFS

La siguiente tabla muestra las versiones de sistema de archivos ZFS disponibles en las versiones de Solaris.

Versión	Solaris 10	Descripción
1	Solaris 10 6/06	Versión inicial de sistemas de archivos ZFS

Versión	Solaris 10	Descripción
2	Solaris 10 10/08	Entradas de directorio mejoradas
3	Solaris 10 10/08	Sin distinción de mayús-minús e identificador exclusivo de sistema de archivo (FUID)
4	Solaris 10 10/09	Propiedades `userquota` y `groupquota`

Índice

B

C

D

E

F

G

H

I

P

R

S

T

U

V

X

Z