Oracle® Solaris ZFS-Administrationshandbuch

Teilenr.: E38889–03
Juni 2013

ORACLE

130704@25097

Inhalt

Vorwort

Das *Oracle Solaris ZFS-Administrationshandbuch* enthält Informationen zum Einrichten und Verwalten von Oracle Solaris ZFS-Dateisystemen.

Dieses Handbuch enthält Informationen für SPARC- und x86-basierte Systeme.

Hinweis – Diese Oracle Solaris-Version unterstützt Systeme auf der Basis der Prozessorarchitekturen SPARC und x86. Die unterstützten Systeme werden in der *Oracle Solaris Hardware Compatibility List* unter http://www.oracle.com/webfolder/technetwork/hcl/index.html angezeigt. Eventuelle Implementierungsunterschiede zwischen den Plattformtypen sind in diesem Dokument angegeben.

Zielgruppe dieses Handbuchs

Dieses Handbuch richtet sich an alle Personen, die daran interessiert sind, Oracle Solaris ZFS-Dateisysteme einzurichten und zu verwalten. Erfahrungen in der Verwendung des Oracle Solaris Betriebssystems oder einer weiteren UNIX-Version sollten vorhanden sein.

Der Aufbau dieses Buches

Die folgende Tabelle erläutert die Kapitel dieses Handbuchs.

Kapitel	Beschreibung
Kapitel 1, „Oracle Solaris ZFS-Dateisystem (Einführung)“	Enthält einen Überblick über ZFS und seine Funktionen bzw. Vorteile. Hier werden auch einige Grundkonzepte sowie Begriffe behandelt.
Kapitel 2, „Erste Schritte mit Oracle Solaris ZFS“	Enthält eine schrittweise Anleitung zum Einrichten einfacher ZFS-Konfigurationen mit einfachen Pools und Dateisystemen. Dieses Kapitel enthält darüber hinaus Informationen zu Hardware und Software, die zum Erstellen von ZFS-Dateisysteme erforderlich ist.

Kapitel	Beschreibung
Kapitel 3, „Verwalten von Oracle Solaris ZFS-Speicher-Pools“	Enthält eine ausführliche Beschreibung zur Erstellung und Administration von ZFS-Speicher-Pools.
Kapitel 4, „Installieren und Booten eines Oracle Solaris ZFS-Root-Dateisystems“	Stellt dar, wie ein ZFS-Dateisystem installiert und gebootet wird. Darüber hinaus wird die Migration eines ZFS-Root-Dateisystems in ein ZFS-Dateisystem mithilfe von Oracle Solaris Live Upgrade behandelt.
Kapitel 5, „Verwalten von Oracle Solaris ZFS-Dateisystemen“	Enthält ausführliche Informationen zum Verwalten von ZFS-Dateisystemen. Hier werden Konzepte wie die hierarchische Dateisystemstrukturierung, Eigenschaftsvererbung, die automatische Administration von Einhängepunkten sowie die Interaktion zwischen Netzwerkdateisystemen behandelt.
Kapitel 6, „Arbeiten mit Oracle Solaris ZFS-Schnappschüssen und -Klonen“	Enthält Informationen zum Erstellen und Verwalten von ZFS-Schnappschüssen und -Klonen.
Kapitel 7, „Schützen von Oracle Solaris ZFS-Dateien mit Zugriffskontrolllisten und Attributen“	Enthält Informationen zum Arbeiten mit Zugriffskontrolllisten. Solche Listen schützen ZFS-Dateien, indem im Vergleich zu den UNIX-Standardzugriffsrechten feiner abgestimmte Zugriffsrechte verwendet werden.
Kapitel 8, „Delegierte Oracle Solaris ZFS-Administration“	Erläutert, wie Benutzern ohne ausreichende Zugriffsrechte mithilfe der delegierten ZFS-Administration die Durchführung von ZFS-Administrationsaufgaben ermöglicht werden kann.
Kapitel 9, „Fortgeschrittene Oracle Solaris ZFS-Themen“	Enthält Informationen zur Verwendung von ZFS-Volumes, von ZFS auf Oracle Solaris-Systemen mit installierten Zonen sowie alternativen Root-Pools.
Kapitel 10, „Problembehebung und Pool-Wiederherstellung in Oracle Solaris ZFS“	Enthält Informationen zum Erkennen von ZFS-Fehlern und zur Wiederherstellung des Normalzustands nach Auftreten dieser Fehler. Darüber hinaus werden hier auch Maßnahmen zum Vermeiden von Fehlfunktionen beschrieben.
Kapitel 11, „Empfohlene Oracle Solaris ZFS-Vorgehensweisen“	Beschreibt die empfohlene Vorgehensweise zum Erstellen, Überwachen und Verwalten der ZFS-Speicherpools und Dateisysteme.
Anhang A, „Oracle Solaris ZFS-Versionsbeschreibungen“	Beschreibung der verfügbaren ZFS-Versionen, der Funktionen jeder Version und des Solaris-Betriebssystems, das die ZFS-Version und die Funktionen bereitstellt.

Verwandte Dokumentation

Die folgenden Handbücher enthalten Informationen zu allgemeinen Themen der Oracle Solaris-Systemadministration:

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

Kontakt zum Oracle Support

Oracle-Kunden können über My Oracle Support den Onlinesupport nutzen. Informationen dazu erhalten Sie unter `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` oder unter `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` (für Hörgeschädigte).

Typografische Konventionen

In der folgenden Tabelle sind die in diesem Handbuch verwendeten typografischen Konventionen aufgeführt.

TABELLE P–1 Typografische Konventionen

Schriftart	Beschreibung	Beispiel
`AaBbCc123`	Namen von Befehlen, Dateien, Verzeichnissen sowie Bildschirmausgaben	Bearbeiten Sie Ihre `.login`-Datei. Verwenden Sie `ls -a`, um eine Liste aller Dateien zu erhalten. `machine_name% Sie haben eine neue Nachricht.`
`AaBbCc123`	Von Ihnen eingegebene Zeichen (im Gegensatz zu auf dem Bildschirm angezeigten Zeichen)	`machine_name%` **`su`** `Passwort:`
aabbcc123	Platzhalter: durch einen tatsächlichen Namen oder Wert zu ersetzen	Der Befehl zum Entfernen einer Datei lautet `rm` *filename*.
AaBbCc123	Buchtitel, neue Ausdrücke; hervorgehobene Begriffe	Lesen Sie hierzu Kapitel 6 im *Benutzerhandbuch*. Ein *Cache* ist eine lokal gespeicherte Kopie. Diese Datei *nicht* speichern. **Hinweis:** Einige hervorgehobene Begriffe werden online fett dargestellt.

Shell-Eingabeaufforderungen in Befehlsbeispielen

Die folgende Tabelle zeigt die UNIX-Standardeingabeaufforderung und die Superuser-Eingabeaufforderung für Shells, die zum Betriebssystem Oracle Solaris gehören. In Befehlbeispielen zeigen die Shell-Eingabeaufforderungen an, ob der Befehl von einem regulären Benutzer oder einem Benutzer mit bestimmten Berechtigungen ausgeführt werden sollte.

TABELLE P–2 Shell-Eingabeaufforderungen

Shell	Eingabeaufforderung
Bash-Shell, Korn-Shell und Bourne-Shell	`$`
Bash-Shell, Korn-Shell und Bourne-Shell für Superuser	`#`
C-Shell	`machine_name%`
C-Shell für Superuser	`machine_name#`

◆ ◆ ◆ KAPITEL 1

1 Oracle Solaris ZFS-Dateisystem (Einführung)

Dieses Kapitel bietet einen Überblick über das Oracle Solaris ZFS-Dateisystem sowie seine Funktionen und Vorteile. Darüber hinaus werden die in diesem Handbuch verwendeten Grundbegriffe erläutert.

Dieses Kapitel enthält die folgenden Abschnitte:

Neuerungen in ZFS

In diesem Abschnitt sind die neuen Leistungsmerkmale des ZFS-Dateisystems zusammengefasst.

- „Unterstützung für ZFS- und Flash-Installation “ auf Seite 24
- „Webbasierte ZFS-Administration“ auf Seite 25

Verbesserungen bei Verwendung des ZFS-Befehls

Oracle Solaris 10 1/13: Die Befehle `zfs` und `zpool` verfügen über einen Unterbefehl `help` , mit dem Sie weitere Informationen über die Unterbefehle `zfs` und `zpool` und die von ihnen unterstützten Optionen aufrufen können. Beispiel:

```
# zfs help
The following commands are supported:
allow       clone       create      destroy     diff        get
groupspace  help        hold        holds       inherit     list
mount       promote     receive     release     rename      rollback
send        set         share       snapshot    unallow     unmount
unshare     upgrade     userspace
For more info, run: zfs help <command>
# zfs help create
usage:
        create [-p] [-o property=value] ... <filesystem>
        create [-ps] [-b blocksize] [-o property=value] ... -V <size> <volume>

# zpool help
The following commands are supported:
add      attach   clear    create   destroy  detach   export   get
help     history  import   iostat   list     offline  online   remove
replace  scrub    set      split    status   upgrade
For more info, run: zpool help <command>
# zpool help attach
usage:
        attach [-f] <pool> <device> <new-device>
```

Weitere Informationen finden Sie in `zfs`(1M) und `zpool`(1M).

Verbesserungen bei ZFS-Schnappschüssen

Oracle Solaris 10 1/13: Diese Version enthält die folgenden Verbesserungen bei ZFS-Schnappschüssen:

- Der Befehl `zfs snapshot` verfügt über einen Alias `snap`, der eine abgekürzte Syntax für diesen Befehl bereitstellt. Beispiel:

  ```
  # zfs snap -r users/home@snap1
  ```

- Der Befehl `zfs diff` stellt eine Aufzählungsoption bereit, `-e`, mit der alle Dateien identifiziert werden können, die zwischen den beiden Schnappschüssen hinzugefügt oder geändert wurden. Die generierte Ausgabe identifiziert alle hinzugefügten Dateien, gibt jedoch keine Löschvorgänge an. Beispiel:

  ```
  # zfs diff -e tank/cindy@yesterday tank/cindy@now
  +       /tank/cindy/
  +       /tank/cindy/file.1
  ```

Sie können auch die Option `-o` verwenden, um gewählte Felder zu identifizieren, die angezeigt werden müssen. Beispiel:

```
# zfs diff -e -o size -o name tank/cindy@yesterday tank/cindy@now
+       7       /tank/cindy/
+       206695  /tank/cindy/file.1
```

Weitere Informationen zum Erstellen von ZFS-Schnappschüssen finden Sie in Kapitel 6, „Arbeiten mit Oracle Solaris ZFS-Schnappschüssen und -Klonen“.

Verbesserte Eigenschaft `aclmode`

Oracle Solaris 10 1/13: Die Eigenschaft `aclmode` ändert das Verhalten von Zugriffskontrolllisten (ACL), wenn ACL-Zugriffsrechte für eine Datei während eines `chmod`-Vorgangs geändert werden. Die Eigenschaft `aclmode` wurde erneut mit den folgenden Eigenschaftswerten eingeführt:

- `discard` – Ein Dateisystem mit einer `aclmode`-Eigenschaft `discard` löscht alle Zugriffskontrolllisteneinträge, die den Modus der Datei nicht darstellen. Dies ist der Standardwert.
- `mask` – Ein Dateisystem mit einer `aclmode` -Eigenschaft `mask` kürzt die Benutzer- oder Gruppenzugriffsrechte. Die Zugriffsrechte werden gekürzt, sodass sie nicht größer sind als die Gruppenzugriffsrechte-Bits, es sei denn, es handelt sich um einen Benutzereintrag, der dieselbe UID wie der Eigentümer der Datei oder des Verzeichnisses hat. In diesem Fall werden die Zugriffsrechte der Zugriffskontrollliste gekürzt, sodass sie nicht größer sind als die Eigentümerzugriffsrechte-Bits. Der Maskenwert behält die Zugriffskontrollliste auch über Modusänderungen hinweg bei, vorausgesetzt, es wurde kein expliziter Vorgang zur Festlegung der Zugriffskontrollliste ausgeführt.
- `passthrough` – Ein Dateisystem mit einer `aclmode`-Eigenschaft `passthrough` gibt an, dass keine Änderungen an der Zugriffskontrollliste vorgenommen werden, außer der Generierung der erforderlichen Zugriffskontrolllisteneinträge zur Darstellung des neuen Modus der Datei oder des Verzeichnisses.

Weitere Informationen finden Sie in Beispiel 7–13.

Oracle Solaris ZFS-Installationsfunktionen

Oracle Solaris 10 8/11: In dieser Version stehen folgende neue Installationsfunktionen zur Verfügung:

- Sie können ein System mit einem ZFS-Flash-Archiv unter Verwendung einer Textmodus-Installation installieren. Weitere Informationen finden Sie in Beispiel 4–3.
- Sie können den Oracle Solaris Live Upgrade-Befehl `luupgrade` verwenden, um ein ZFS-Root-Flash-Archiv zu installieren. Weitere Informationen finden Sie in Beispiel 4–8.

- Mit dem Befehl `lucreate` von Oracle Solaris Live Upgrade können Sie ein separates /var-Dateisystem angeben. Weitere Informationen finden Sie in Beispiel 4–5.

ZFS send-Datenstrom-Verbesserungen

Oracle Solaris 10 8/11: In dieser Version können Sie Dateisystemeigenschaften festlegen, die in einem Schnappschuss-Datenstrom gesendet und empfangen werden. Diese Verbesserungen sorgen für Flexibilität beim Übernehmen der Dateisystemeigenschaften in einem send-Datenstrom für das empfangene Zieldateisystem oder beim Bestimmen, ob die lokalen Dateisystemeigenschaften, wie z. B. der Eigenschaftswert `mountpoint`, beim Empfang ignoriert werden sollen.

Weitere Informationen finden Sie unter „Anwenden verschiedener Eigenschaftswerte auf einen ZFS-Schnappschussdatenstrom“ auf Seite 234.

ZFS-Schnappschussunterschiede (`zfs diff`)

Oracle Solaris 10 8/11: In dieser Version können Sie mithilfe des Befehls `zfs diff` ZFS-Schnappschussunterschiede bestimmen.

Beispielsweise werden folgende zwei Schnappschüsse erstellt:

```
$ ls /tank/cindy
fileA
$ zfs snapshot tank/cindy@0913
$ ls /tank/cindy
fileA  fileB
$ zfs snapshot tank/cindy@0914
```

Um beispielsweise die Unterschiede zwischen zwei Schnappschüssen zu ermitteln, verwenden Sie folgende Syntax:

```
$ zfs diff tank/cindy@0913 tank/cindy@0914
M       /tank/cindy/
+       /tank/cindy/fileB
```

In der Ausgabe gibt `M` an, dass das Verzeichnis geändert wurde. Das `+` gibt an, dass `fileB` im späteren Schnappschuss vorhanden ist.

Weitere Informationen finden Sie unter „Ermitteln von ZFS-Schnappschussunterschieden (`zfs diff`)“ auf Seite 224.

Wiederherstellung von ZFS-Speicher-Pools und Verbesserung der Systemleistung

Oracle Solaris 10 8/11: In dieser Version stehen folgende neue Funktionen des ZFS-Speicherpools zur Verfügung:

- Mit dem Befehl `zpool import -m` können Sie einen Pool mit einem fehlenden Protokoll importieren. Weitere Informationen finden Sie unter „Importieren eines Pools mit fehlendem Protokolliergerät“ auf Seite 105.
- Sie können einen Pool im schreibgeschützten Modus importieren. Diese Funktion dient primär der Pool-Wiederherstellung. Wenn nicht auf einen beschädigten Pool zugegriffen werden kann, weil die zugrunde liegenden Geräte beschädigt sind, können Sie den Pool im schreibgeschützten Modus importieren, um die Daten wiederherzustellen. Weitere Informationen finden Sie unter „Importieren eines Pools im schreibgeschützten Modus“ auf Seite 106.
- Bei einem RAID-Z-Speicherpool (`raidz1`, `raidz2` oder `raidz3`), der in dieser Version erstellt wird, werden bestimmte latenzabhängige Metadaten automatisch gespiegelt, um die Leistung des E/A-Lesedurchsatzes zu verbessern. Bei vorhandenen RAID-Z-Pools, die auf Pool-Version 29 oder höher aktualisiert werden, erfolgt eine Spiegelung bestimmter Metadaten für alle neu geschriebenen Daten.

  Gespiegelte Metadaten in einem RAID-Z-Pool bieten *keinen* zusätzlichen Schutz vor Hardwareausfällen, wie ihn beispielsweise ein gespiegelter Speicher-Pool liefern würde. Gespiegelte Metadaten nehmen zwar zusätzlichen Speicherplatz in Anspruch, der RAID-Z-Schutz bleibt jedoch derselbe wie in den früheren Versionen. Diese Verbesserung dient ausschließlich der Leistungssteigerung.

Anpassen des synchronen ZFS-Verhaltens

Solaris 10 8/11: In dieser Version können Sie mithilfe der Eigenschaft `sync` das synchrone Verhalten eines ZFS-Dateisystems bestimmen.

Beim standardmäßigen synchronen Verhalten werden alle synchronen Dateisystem-Transaktionen in das Intent-Protokoll geschrieben und alle Geräte entfernt, damit die Daten sicher sind. Die Deaktivierung des standardmäßigen synchronen Verhaltens wird nicht empfohlen. Anwendungen, die von der Unterstützung des synchronen Verhaltens abhängen, können beschädigt werden, sodass es zu Datenverlust kommen könnte.

Die Eigenschaft `sync` kann vor oder nach Erstellung des Dateisystems festgelegt werden. In beiden Fällen wird der Eigenschaftswert sofort übernommen. Beispiel:

```
# zfs set sync=always tank/neil
```

Der Parameter `zil_disable` steht in Oracle Solaris-Versionen mit der Eigenschaft `sync` nicht mehr zur Verfügung.

Weitere Informationen finden Sie in Tabelle 5–1.

Verbesserte ZFS-Pool-Nachrichten

Oracle Solaris 10 8/11: In dieser Version können Sie mit der Option `-T` ein Intervall und einen Zählerwert für die Befehle `zpool list` und `zpool status` bereitstellen, um zusätzliche Informationen anzuzeigen.

Darüber hinaus stellt der Befehl `zpool status` folgende weitere Informationen zu Pool-Scrub und Spiegelung bereit:

- Bericht, während die Spiegelung läuft. Beispiel:

  ```
  scan: resilver in progress since Thu Jun  7 14:41:11 2012
      3.83G scanned out of 73.3G at 106M/s, 0h11m to go
      3.80G resilvered, 5.22% done
  ```

- Bericht, während die Bereinigung läuft. Beispiel:

  ```
  scan: scrub in progress since Thu Jun  7 14:59:25 2012
      1.95G scanned out of 73.3G at 118M/s, 0h10m to go
      0 repaired, 2.66% done
  ```

- Meldung, dass die Spiegelung abgeschlossen ist. Beispiel:

  ```
  resilvered 73.3G in 0h13m with 0 errors on Thu Jun  7 14:54:16 2012
  ```

- Meldung, dass die Bereinigung abgeschlossen ist. Beispiel:

  ```
  scan: scrub repaired 512B in 1h2m with 0 errors on Thu Jun  7 15:10:32 2012
  ```

- Meldung, dass die laufende Bereinigung abgebrochen wurde. Beispiel:

  ```
  scan: scrub canceled on Thu Jun  7 15:19:20 MDT 2012
  ```

- Meldungen zum Abschluss von Bereinigung und Spiegelung bleiben auch nach Systemneustarts erhalten.

In der folgenden Syntax wird die Zeitintervall- und Zählparameteroption verwendet, um Informationen zur laufenden Pool-Spiegelung anzuzeigen. Sie können die Informationen mit dem `-T d`-Wert im standardmäßigen Datumsformat oder mit dem `-T u`-Wert in einem internen Format anzeigen.

```
# zpool status -T d tank 3 2
Wed Nov 14 15:44:34 MST 2012
  pool: tank
 state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Wed Nov 14 15:44:34 2012
    2.96G scanned out of 4.19G at 189M/s, 0h0m to go
    1.48G resilvered, 70.60% done
config:
```

```
        NAME                       STATE     READ WRITE CKSUM
        tank                       DEGRADED     0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 DEGRADED     0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  DEGRADED     0     0     0  (resilvering)

errors: No known data errors
```

Interoperabilitätsverbesserungen für ZFS-Zugriffskontrolllisten

Oracle Solaris 10 8/11: In dieser Version werden folgende Verbesserungen für Zugriffskontrolllisten bereitgestellt:

- Einfache Zugriffskontrolllisten erfordern keine `deny`-Zugriffskontrolleinträge (ACEs), außer bei ungewöhnlichen Berechtigungen. Beispiel: Ein Modus `0644`, `0755` oder `0664` erfordert keine `deny`ACEs, ein Modus wie `0705`, `0060` usw. erfordert hingegen `deny`-ACEs.

  Das alte Verhalten umfasst `deny`-ACEs in einer gewöhnlichen ACL, wie `644`. Beispiel:

  ```
  # ls -v file.1
  -rw-r--r--   1 root     root      206663 Jun 14 11:52 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
  ```

  Das neue Verhalten für eine gewöhnliche Zugriffskontrollliste wie `644` schließt die `deny`-ACEs nicht mehr mit ein. Beispiel:

  ```
  # ls -v file.1
  -rw-r--r--   1 root     root      206663 Jun 22 14:30 file.1
       0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
           /read_attributes/write_attributes/read_acl/write_acl/write_owner
           /synchronize:allow
       1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
       2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
           :allow
  ```

- Während der Vererbung werden Zugriffskontrolllisten nicht mehr in mehrere Zugriffskontrolleinträge aufgeteilt, um die ursprüngliche unveränderte Berechtigung zu erhalten. Stattdessen werden die Berechtigungen nach Bedarf geändert, um den Dateierstellungsmodus durchzusetzen.

- Das Verhalten der Eigenschaft `aclinherit` beinhaltet eine Reduzierung der Berechtigungen, wenn die Eigenschaft auf `restricted` gesetzt wird. Dies hat zur Folge, dass die Zugriffskontrolllisten während der Vererbung nicht mehr in mehrere Zugriffskontrolleinträge aufgeteilt werden.
- Eine neue Regel für die Berechnung des Berechtigungsmodus gibt an, dass wenn eine Zugriffskontrollliste einen *user*-Zugriffskontrolleintrag enthält und dieser Benutzer gleichzeitig der Dateieigentümer ist, diese Berechtigungen in der Berechtigungsmodus-Berechnung inbegriffen sind. Die gleiche Regel gilt, wenn ein *group*-Zugriffskontrolleintrag der Gruppeneigentümer dieser Datei ist.

Weitere Informationen finden Sie in Kapitel 7, „Schützen von Oracle Solaris ZFS-Dateien mit Zugriffskontrolllisten und Attributen“.

Teilung eines ZFS-Speicher-Pools mit Datenspiegelung (`zpool split`)

Oracle Solaris 10 9/10: In dieser Version können Sie mit dem Befehl `zpool split` einen gespiegelten Speicherpool teilen, wodurch dem gespiegelten Pool eine oder mehrere Festplatten entnommen werden, um einen weiteren identischen Pool anzulegen.

Weitere Informationen finden Sie unter „Erstellen eines neuen Pools durch Teilen eines ZFS-Speicher-Pools mit Datenspiegelung“ auf Seite 71.

Neuer ZFS-Systemprozess

Oracle Solaris 10 9/10: In dieser Version ist jedem ZFS-Speicherpool ein Prozess (`zpool-`*poolname*) zugeordnet. Die Teilprozesse dieses Prozesses dienen zur Verarbeitung von E/A-Vorgängen, die den Pool betreffen, wie beispielsweise Komprimierung und Prüfsummenbildung. Dieser Prozess soll Aufschluss über die CPU-Auslastung der einzelnen Speicher-Pools geben.

Informationen zu diesen laufenden Prozessen können mithilfe der Befehle `ps` und `prstat` angezeigt werden. Diese Prozesse stehen nur in der globalen Zone zur Verfügung. Weitere Informationen finden Sie unter `SDC(7)`.

Verbesserungen für den Austausch von ZFS-Speichergeräten

Oracle Solaris 10 9/10: In dieser Version wird ein Systemereignis oder *sysevent* bereitgestellt, wenn die Festplatten in einem Pool durch größere Festplatten ersetzt werden. ZFS wurde verbessert, um diese Ereignisse zu erkennen, und passt den Pool basierend auf der neuen Größe

der Festplatte an, wobei die Einstellung der Eigenschaft `autoexpand` berücksichtigt wird. Mit der Pool-Eigenschaft `autoexpand` können Sie die automatische Pool-Erweiterung aktivieren oder deaktivieren, wenn eine kleinere Festplatte durch eine größere ersetzt wird.

Dank dieser Verbesserungen können Sie die Pool-Größe erweitern, ohne einen Pool zu exportieren und zu importieren oder das System neu zu starten.

Die automatische Erweiterung der LU-Nummer ist beispielsweise für den Pool `tank` aktiviert.

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

Sie können auch einen Pool erstellen, dessen Eigenschaft `autoexpand` aktiviert ist.

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

Die Eigenschaft `autoexpand` ist standardmäßig deaktiviert, sodass Sie festlegen können, ob die Pool-Größe beim Ersetzen einer kleineren Festplatte durch eine größere erweitert werden soll.

Die Pool-Größe kann außerdem mithilfe des Befehls `zpool online -e` erweitert werden. Beispiel:

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

Außerdem können Sie, nachdem eine größere Festplatte eingebunden oder verfügbar gemacht wurde, die Eigenschaft `autoexpand` zurücksetzen, indem Sie den Befehl `zpool replace` verwenden. Der folgende Pool wird beispielsweise mit einer 8-GB-Festplatte (`c0t0d0`) erstellt. Die 8-GB-Festplatte wird durch eine 16-GB-Festplatte (`c1t13d0`) ersetzt, aber die Pool-Kapazität wird erst dann erweitert, wenn die Eigenschaft `autoexpand` aktiviert ist.

```
# 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  -
```

Eine weitere Möglichkeit, die Festplatte ohne Aktivierung der Eigenschaft `autoexpand` zu erweitern, ist die Verwendung des Befehls `zpool online -e`. Der Befehl kann auch dann ausgeführt werden, wenn das Gerät bereits in Betrieb genommen ist. Beispiel:

```
# 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  -
```

Weitere Verbesserungen für den Austausch von Speichergeräten in dieser Version sind folgende:

- In früheren Versionen war ZFS nicht fähig, eine vorhandene Festplatte durch eine andere Festplatte zu ersetzen oder eine Festplatte einzubinden, wenn die Ersatzfestplatte eine geringfügig unterschiedliche Kapazität hatte. In dieser Version können Sie eine vorhandene Festplatte durch eine andere Festplatte ersetzen oder eine neue Festplatte mit nahezu derselben Kapazität einbinden, unter der Voraussetzung, dass der Pool nicht bereits voll ist.
- In dieser Version müssen Sie das System nicht neu starten oder einen Pool exportieren und importieren, um die Pool-Größe zu erweitern. Wie zuvor beschrieben können Sie die Eigenschaft `autoexpand` aktivieren oder den Befehl `zpool online -e` verwenden, um die Pool-Größe zu erweitern.

Weitere Informationen zum Austauschen von Geräten finden Sie unter „Austauschen von Geräten in einem Speicher-Pool“ auf Seite 77.

Unterstützung für ZFS- und Flash-Installation

Solaris 10 10/09: In dieser Version haben Sie die Möglichkeit, ein JumpStart-Profil zur Identifizierung eines Flash-Archivs eines ZFS-Root-Pools einzurichten. Weitere Informationen finden Sie unter „Installieren eines ZFS-Root-Dateisystems (Oracle Solaris Flash-Archiv-Installation)“ auf Seite 124.

Zonenmigration in einer ZFS-Umgebung

Solaris 10 5/09: Diese Version bietet eine erweiterte Unterstützung für das Migrieren von Zonen in einer ZFS-Umgebung mit Oracle Solaris Live Upgrade. Weitere Informationen finden Sie unter „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147.

Die Solaris 10 5/09-Versionshinweise enthalten eine Liste der bekannten Probleme mit diesem Release.

Unterstützung für Installation und Booten von ZFS-Root-Dateisystemen

Solaris 10 10/08: Mit dieser Solaris-Version können ZFS-Root-Dateisysteme installiert und gebootet werden. Das Installieren eines ZFS-Root-Dateisystems ist sowohl mit der Erstinstallationsoption als auch mit der JumpStart-Funktion möglich. Sie können aber auch Oracle Solaris Live Upgrade verwenden, um ein UFS-Root-Dateisystem auf ein ZFS-Root-Dateisystem zu migrieren. Außerdem steht ZFS-Unterstützung für Swap- und Dump-Geräte zur Verfügung. Weitere Informationen finden Sie in Kapitel 4, „Installieren und Booten eines Oracle Solaris ZFS-Root-Dateisystems“.

Die Solaris 10 10/08-Versionshinweise enthalten eine Liste der bekannten Probleme mit diesem Release.

Webbasierte ZFS-Administration

Solaris 10 6/06: Mit einem webbasierten ZFS-Administrationstool, der ZFS-Administrationskonsole, können Sie folgende Administrationsaufgaben ausführen:

- Erstellen eines neuen Datenspeicher-Pools
- Erweitern der Kapazität eines vorhandenen Datenspeicher-Pools
- Verlagern (Exportieren) eines Datenspeicher-Pools auf ein anderes System
- Importieren eines zuvor exportierten Datenspeicher-Pools, um diesen auf einem anderen System verfügbar zu machen
- Anzeigen von Informationen zu Datenspeicher-Pools
- Erstellen von Dateisystemen
- Erstellen von Volumes
- Erstellen eines Schnappschusses eines Dateisystems oder Volumes
- Wiederherstellen eines früheren Schnappschusses eines Dateisystems

Sie können mithilfe eines sicheren Webbrowsers unter der folgenden URL auf die ZFS-Administrationskonsole zugreifen:

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

Wenn Sie die entsprechende URL eingeben und die ZFS-Administrationskonsole nicht erreichen, kann es sein, dass deren Server nicht läuft. Geben Sie den folgenden Befehl ein, um den Server zu starten:

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

Geben Sie den folgenden Befehl ein, wenn der Server beim Hochfahren des Systems automatisch gestartet werden soll:

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

Hinweis – Die Solaris Management Console (`smc`) kann nicht zur Administration von ZFS-Speicher-Pools bzw. -Dateisystemen verwendet werden.

Was ist Oracle Solaris ZFS?

Das ZFS-Dateisystem ist ein Dateisystem, das die Administration von Dateisystemen grundlegend ändert. Es besitzt Leistungsmerkmale und Vorteile, die in keinem anderen heutzutage verfügbaren Dateisystem zu finden sind. ZFS ist robust, skalierbar und einfach zu verwalten.

Speicherpools in ZFS

Die physische Datenspeicherung beruht bei ZFS auf dem Konzept der *Speicher-Pools*. Früher wurden Dateisysteme auf ein einziges physisches Datenspeichergerät aufsetzend konzipiert. Damit mehrere Datenspeichergeräte adressiert werden können und Datenredundanz möglich wird, wurde das Konzept eines so genannten *Volume Manager* entwickelt, um für die Darstellung eines einzelnen Datenspeichergeräts zu sorgen. Dadurch müssen Dateisysteme nicht modifiziert werden, wenn mehrere Datenspeichergeräte genutzt werden sollen. Dieses Konzept brachte mehr Komplexität mit sich und behinderte die Weiterentwicklung, da das Dateisystem keine Kontrolle über die physische Speicherung von Daten auf den virtualisierten Volumes hatte.

In ZFS ist die Administration einzelner Volumes nicht mehr erforderlich. Anstatt einer erzwungenen Erstellung virtueller Volume-Systeme fasst ZFS Datenspeichergeräte in so genannten Speicher-Pools zusammen. Ein solcher Speicher-Pool bestimmt die physischen Eigenschaften der Speicherung (Gerätestruktur, Datenredundanz usw.) und fungiert als flexibler Datenspeicher, aus dem Dateisysteme erstellt werden können. Dateisysteme sind nicht mehr auf bestimmte Speichergeräte beschränkt und können die Festplattenkapazität im Pool gemeinsam nutzen. Sie müssen die Kapazität eines Dateisystems nicht mehr vorher festlegen, da die Dateisysteme automatisch innerhalb der Festplattenkapazität erweitert werden, die im Speicher-Pool verfügbar ist. Wenn ein Pool um neuen Speicherplatz erweitert wird, können alle Dateisysteme dieses Pools diese neue Festplattenkapazität sofort nutzen, ohne dass dafür Konfigurationen geändert werden müssen. In mancherlei Hinsicht lässt sich die Funktionsweise eines Speicher-Pools mit der eines virtuellen Speichersystems vergleichen: Wenn ein DIMM-Speicherchip in ein System eingebaut wird, werden Sie vom Betriebssystem nicht gezwungen, Befehle auszuführen, um den neuen Speicher zu konfigurieren und einzelnen Prozessen zuzuweisen. Alle Prozesse des Systems verwenden automatisch diesen zusätzlichen Speicherplatz.

Transaktionale Semantik

ZFS ist ein transaktionales Dateisystem. Das bedeutet, dass der Dateisystemstatus auf dem Datenträger stets konsistent ist. Bei herkömmlichen Dateisystemen werden vorhandene Daten überschrieben. Dies kann dazu führen, dass sich ein Dateisystem in einem instabilen Zustand befindet, wenn beispielsweise zwischen dem Zeitpunkt der Zuweisung eines Datenblocks und dem der Verknüpfung mit einem Verzeichnis ein Stromausfall aufgetreten war. Früher wurde dieses Problem durch den Befehl `fsck` behoben. Dieser Befehl diente dem Zweck, den Zustand des Dateisystems zu überprüfen und zu versuchen, Inkonsistenzen zu beheben. Das Problem von inkonsistenten Dateisystemen verursachte Systemadministratoren viele Unannehmlichkeiten, und der Befehl `fsck` bot nicht die Garantie, alle Probleme zu beheben. Zuletzt wurden Dateisysteme entwickelt, die auf dem Konzept des so genannten *Journaling* beruhen. Beim Journaling werden Aktionen in einem eigenen "Journal" festgehalten, das im Falle von Systemabstürzen sicher *eingespielt* werden kann. Diese Methode verursacht jedoch unnötigen Datenverarbeitungsaufwand, da Daten zweimal geschrieben werden müssen. Außerdem entstehen oft weitere Probleme, wenn beispielsweise das Journal nicht fehlerfrei gelesen werden konnte.

Transaktionale Dateisysteme verwalten Daten mithilfe der so genannten *Copy on Write*-Semantik. Bei diesem Verfahren werden niemals Daten überschrieben, und jegliche Folge von Vorgängen wird entweder vollständig festgeschrieben oder ganz ignoriert. Das bedeutet, dass ein Dateisystem bei diesem Verfahren durch Stromausfälle oder Systemabstürze grundsätzlich nicht beschädigt werden kann. Obwohl ganz zuletzt gespeicherte Daten unter Umständen verloren gehen können, bleibt das Dateisystem selbst stets konsistent. Darüber hinaus werden (mithilfe des `O_DSYNC`-Flags geschriebene) synchrone Daten stets vor der Rückkehr geschrieben, sodass sie niemals verloren gehen.

Prüfsummen und Daten mit Selbstheilungsfunktion

In ZFS werden alle Daten und Metadaten durch einen vom Benutzer auswählbaren Prüfsummenalgorithmus verifiziert. Herkömmliche Dateisysteme führten die Prüfsummenverifizierung datenblockweise aus, was auf die Volume-Administrationsschicht und den Aufbau dieser Dateisysteme zurückzuführen war. Aufgrund des herkömmlichen Aufbaus dieser Dateisysteme kann es vorkommen, dass bestimmte Fehler wie z. B. das Speichern eines Datenblocks an einem falschen Speicherort dazu führen kann, dass trotz fehlerhafter Daten keine Prüfsummenfehler vorhanden sind. ZFS-Prüfsummen werden so gespeichert, dass solche Fehler erkannt und der ordnungsgemäße Zustand des Dateisystems wiederhergestellt werden kann. die gesamte Prüfsummenverifzierung und Datenwiederherstellung auf der Dateisystemebene durchgeführt werden und für Anwendungen transparent sind.

Darüber hinaus bietet ZFS Selbstheilungsfunktionen für Daten. ZFS unterstützt Speicher-Pools mit unterschiedlichen Stufen der Datenredundanz. Bei Erkennung beschädigter Datenblöcke

ruft ZFS die unbeschädigten Daten von einer redundanten Kopie ab und repariert die beschädigten Daten, indem es diese durch korrekte Daten ersetzt.

Konkurrenzlose Skalierbarkeit

Ein Schlüsselelement des ZFS-Dateisystems ist Skalierbarkeit. Das Dateisystem beruht auf der 128-Bit-Architektur, was die Speicherung von 256 Billiarden Zettabyte Daten ermöglicht. Alle Metadata werden dynamisch zugewiesen. Damit entfällt die Notwendigkeit, Inodes vorher zuweisen bzw. die Skalierbarkeit bei der ersten Erstellung eines Dateisystems anderweitig einschränken zu müssen. Alle Algorithmen wurden im Hinblick auf eine bestmögliche Skalierbarkeit entwickelt. Verzeichnisse können bis zu 2^{48} (256 Billionen) Einträge enthalten, und für die Anzahl der Dateisysteme bzw. die Anzahl der in einem Dateisystem enthaltenen Dateien bestehen keine Beschränkungen.

ZFS-Schnappschüsse

Ein *Schnappschuss* ist eine schreibgeschützte Kopie eines Dateisystems bzw. Volume. Schnappschüsse können schnell und einfach erstellt werden. Anfänglich belegen Schnappschüsse keine zusätzliche Festplattenkapazität im Pool.

Wenn Daten innerhalb des aktiven Datasets geändert werden, belegt der Schnappschuss Festplattenkapazität, da die alten Daten weiterhin referenziert werden. So verhindern Schnappschüsse, dass der von den Daten belegte Speicherplatz für den Pool freigegeben wird.

Vereinfachte Administration

Eine der wichtigsten Eigenschaften von ZFS ist das erheblich vereinfachte Administrationsmodell. Durch hierarchische Dateisystemstrukturen, Eigenschaftsvererbung sowie automatische Administration von Einhängepunkten und NFS-Netzwerksemantik können ZFS-Dateisysteme auf einfache Weise erstellt und verwaltet werden, ohne dass dafür mehrere Befehle ausgeführt oder Konfigurationsdateien bearbeitet werden müssen. Sie können auf einfache Weise Kontingente bzw. Reservierungen vornehmen, Datenkomprimierung aktivieren/deaktivieren oder Einhängepunkte für mehrere Dateisysteme mit einem einzigen Befehl verwalten. Sie können Datenspeichergeräte überprüfen oder reparieren, ohne Kenntnisse über einen separaten Volume-Manager-Befehlssatz zu besitzen. Sie können Datenströme von Dateisystem-Schnappschüsse senden und empfangen.

ZFS verwaltet Dateisysteme über eine Hierarchie, die eine vereinfachte Administration von Eigenschaften wie z. B. Kontingenten, Reservierungen, Komprimierung und Einhängepunkten ermöglicht In diesem Modell werden Dateisysteme zur zentralen Administrationsstelle. Dateisysteme sind sehr kostengünstig (ähnlich wie die Erstellung eines neuen Verzeichnisses).

Deswegen sollten Sie für jeden Benutzer, jedes Projekt, jeden Arbeitsbereich usw. ein neues Dateisystem erstellen. Dieses Konzept ermöglicht Ihnen, Administrationspunkte genau zu definieren.

In ZFS verwendete Begriffe

In diesem Abschnitt werden die im vorliegenden Handbuch verwendeten Begriffe erläutert:

Alternative Boot-Umgebung Eine Boot-Umgebung, die mit dem Befehl `lucreate` erstellt und mit dem Befehl `luupgrade` aktualisiert werden kann. Bei dieser Boot-Umgebung handelt es sich aber weder um die aktive noch um die primäre Boot-Umgebung. Die alternative Boot-Umgebung kann mithilfe des Befehls `luactivate` zur primären Boot-Umgebung gemacht werden.

Checksum Mithilfe eines 256-Bit-Hash-Algorithmus verschlüsselte Daten eines Dateisystemblocks. Prüfsummenalgorithmen reichen vom einfachen und schnellen Fletcher4-Algorithmus (die Standardeinstellung) bis hin zu Hash-Algorithmen mit starker Verschlüsselung wie z. B. SHA256.

Klon Ein Dateisystem, dessen anfänglicher Inhalt identisch mit dem Inhalt eines Schnappschusses ist.

Informationen zu Klonen finden Sie unter „Überblick über ZFS-Klone“ auf Seite 225.

Dataset Ein allgemeiner Name für die folgenden ZFS-Komponenten: Klone, Dateisysteme, Schnappschüsse und Volumes.

Jeder Datensatz wird im ZFS-Namespace durch einen eindeutigen Namen identifiziert. Datasets werden mithilfe des folgenden Formats benannt:

pool/*path*[@*snapshot*]

pool Der Name des Speicher-Pools, der das Dataset enthält

path Ein durch Schrägstriche begrenzter Pfadname für die Dataset-Komponente

snapshot Eine optionale Komponente, die den Snapshot eines Datasets identifiziert

Weitere Informationen zu Datasets finden Sie in Kapitel 5, „Verwalten von Oracle Solaris ZFS-Dateisystemen“.

Dateisystem	Ein ZFS-Datensatz der Art `filesystem`, der im Standard-Namespace des Systems eingehängt ist und sich wie jedes andere Dateisystem verhält. Weitere Informationen zu Dateisystemen finden Sie in Kapitel 5, „Verwalten von Oracle Solaris ZFS-Dateisystemen“.
Spiegelung	Eine Konfiguration mit virtuellen Geräten, in der identische Kopien von Daten auf zwei oder mehr Festplatten gespeichert werden. Wenn eine Festplatte im Datenspiegelungssystem ausfällt, stellt eine andere Festplatte dieses Systems die gleichen Daten bereit.
Pool	Eine logische Gruppe von Geräten, die das Layout und die physischen Merkmale des verfügbaren Speichers beschreibt. Datensätzen wird Festplattenkapazität aus einem Pool zugewiesen. Weitere Informationen zu Speicher-Pools finden Sie in Kapitel 3, „Verwalten von Oracle Solaris ZFS-Speicher-Pools“.
Primäre Boot-Umgebung	Eine Boot-Umgebung, aus der mit dem Befehl `lucreate` die alternative Boot-Umgebung erstellt wird. Standardmäßig ist die primäre Boot-Umgebung die aktuelle Boot-Umgebung. Dieses Standardverhalten kann mit `lucreate` und der Option `-s` außer Kraft gesetzt werden.
RAID-Z	Ein virtuelles Gerät, das Daten und Paritäten auf mehreren Festplatten speichert. Weitere Informationen zu RAID-Z finden Sie unter „Speicher-Pools mit RAID-Z-Konfiguration“ auf Seite 50.
Resilvering	Den Vorgang des Kopierens von Daten von einem Datenspeichergerät auf ein anderes Gerät nennt man *Resilvering*. Wenn beispielsweise ein Spiegelungsgerät ersetzt oder offline geschaltet wird, werden die Daten eines auf dem aktuellen Stand befindlichen Spiegelungsgeräts auf das neu wiederhergestellte Gerät kopiert. Dieser Vorgang heißt in herkömmlichen Datenträgermanagement-Produkten *Neusynchronisierung der Datenspiegelung*. Weitere Informationen zum ZFS-Resilvering finden Sie in „Anzeigen des Resilvering-Status“ auf Seite 316.

Schnappschuss	Eine schreibgeschützte Kopie eines Dateisystems oder Volumes zu einem bestimmten Zeitpunkt. Weitere Informationen zu Snapshots finden Sie in „Überblick über ZFS-Schnappschüsse“ auf Seite 217.
Virtuelles Gerät	Ein logisches Gerät in einem Pool. Dies kann ein physisches Datenspeichergerät, eine Datei oder ein Geräteverbund sein. Weitere Informationen zu virtuellen Geräten finden Sie unter „Anzeigen von Informationen zu virtuellen Geräten in Storage-Pools“ auf Seite 59.
Volume	Ein Dataset, das eine Blockeinheit darstellt. Beispielsweise lässt sich ein ZFS-Volume als Swap-Gerät erstellen. Weitere Informationen zu ZFS-Volumes finden Sie unter „ZFS-Volumes“ auf Seite 283.

Konventionen für das Benennen von ZFS-Komponenten

Jede ZFS-Komponente wie beispielsweise ein Dataset oder ein Pool muss nach den folgenden Regeln benannt werden:

- Eine Komponente darf nur alphanumerische Zeichen sowie die folgenden vier Sonderzeichen enthalten:
  - Unterstrich (_)
  - Bindestrich (-)
  - Doppelpunkt (:)
  - Punkt (.)
- Poolnamen müssen mit einem Buchstaben beginnen und können nur alphanumerische Zeichen sowie Unterstriche (_), Bindestriche (-) und Punkte enthalten (.). Beachten Sie die folgenden Einschränkungen bei Poolnamen:
  - Die Zeichenfolge `c[0-9]` ist am Namensbeginn nicht erlaubt.
  - Der Name `log` ist reserviert.
  - Namen, die mit `mirror`, `raidz` , `raidz1`, `raidz2`, `raidz3` oder `spare` beginnen, sind nicht erlaubt, da diese Namen reserviert sind.
  - Pool-Namen dürfen kein Prozentzeichen (`%`) enthalten.
- Dataset-Namen müssen mit einem alphanumerischen Zeichen beginnen.
- Dataset-Namen dürfen kein Prozentzeichen (`%`) enthalten.

Außerdem sind leere Komponenten unzulässig.

Unterschiede zwischen Oracle Solaris ZFS und herkömmlichen Dateisystemen

Granularität von ZFS-Dateisystemen

Früher waren Dateisysteme auf ein Datenspeichergerät und damit auf die Kapazität dieses Geräts beschränkt. Das Erstellen und Neuerstellen herkömmlicher Dateisysteme aufgrund von Kapazitätsbeschränkungen ist zeitaufwändig und in einigen Fällen kompliziert. Herkömmliche Administrationslösungen unterstützen diesen Prozess.

Da ZFS-Dateisysteme nicht auf spezielle Geräte beschränkt sind, können sie ähnlich wie Verzeichnisse schnell und einfach erstellt werden. ZFS-Dateisysteme werden innerhalb der verfügbaren Festplattenkapazität des Speicher-Pools, in dem sie enthalten sind, automatisch vergrößert.

Anstatt ein Dateisystem (z. B. `/export/home`) zum Verwalten von Benutzerverzeichnissen zu erstellen, können Sie pro Benutzer ein Dateisystem anlegen. Sie können Dateisysteme auf einfache Weise einrichten und verwalten, indem Sie Eigenschaften festlegen, die von untergeordneten Dateisystemen innerhalb einer Hierarchie geerbt werden.

Ein Beispiel, das zeigt, wie eine Dateisystemhierarchie erstellt wird, finden Sie unter „Erstellen einer ZFS-Dateisystemhierarchie“ auf Seite 41.

Berechnung von ZFS-Festplattenkapazität

ZFS beruht auf dem Konzept der Datenspeicherung in Pools. Im Gegensatz zu herkömmlichen, physischen Datenträgern direkt zugewiesenen Dateisystemen teilen sich ZFS-Dateisysteme in einem Pool den in diesem Pool verfügbaren Speicherplatz. Somit kann sich die verfügbare Festplattenkapazität, die von Serviceprogrammen wie z. B. `df` gemeldet wird, auch dann ändern, wenn das betreffende Dateisystem inaktiv ist, da andere Dateisysteme im Pool Festplattenkapazität belegen bzw. freigeben.

Bitte beachten Sie, dass die maximal mögliche Dateisystemkapazität durch die Zuteilung von Kontingenten beschränkt werden kann. Weitere Informationen zu Kontingenten finden Sie unter „Setzen von Kontingenten für ZFS-Dateisysteme“ auf Seite 211. Durch Reservierungen

kann einem Dateisystem eine bestimmte Festplattenkapazität garantiert werden. Weitere Informationen zu Reservierungen finden Sie unter „Setzen von Reservierungen für ZFS-Dateisysteme“ auf Seite 214. Dieses Modell gleicht dem NFS-Modell sehr; dort werden mehrere Verzeichnisse vom gleichen Dateisystem (z. B. `/home`) eingehängt.

Alle Metadaten werden in ZFS dynamisch zugewiesen. In herkömmlichen Dateisystemen erfolgt die Zuweisung von Metadaten meist vorher. Deswegen muss für diese Metadaten schon beim Erstellen des Dateisystems zusätzlicher Speicherplatz reserviert werden. Dieses Verhalten bedeutet weiterhin, dass die mögliche Gesamtdateianzahl eines Dateisystems schon vorher festgelegt ist. Da ZFS Metadaten nach Bedarf zuweist, muss vorher kein zusätzlicher Speicherplatz reserviert werden, und die Anzahl von Dateien wird lediglich durch die verfügbare Festplattenkapazität begrenzt. Die Ausgabe des Befehls `df -g` ist bei ZFS anders als bei herkömmlichen Dateisystemen zu interpretieren. Die unter `total files` ausgegebene Dateinanzahl ist lediglich ein Schätzwert, der auf dem Betrag des im Pool verfügbaren Speicherplatzes beruht.

ZFS ist ein transaktionales Dateisystem. Das bedeutet, dass die meisten Dateisystemmodifikationen in Transaktionsgruppen zusammengefasst und asynchron auf dem Datenträger ausgeführt werden. Diese Modifikationen gelten so lange als *anstehende Änderungen*, bis sie auf dem Datenträger festgeschrieben sind. Die von einer Datei oder einem Dateisystem belegte, verfügbare und referenzierte Festplattenkapazität berücksichtigt keine anstehenden Änderungen. Anstehende Änderungen werden im Allgemeinen innerhalb weniger Sekunden abgeschlossen. Auch nach dem Festschreiben einer Änderung auf der Festplatte durch `fsync(3c)` oder `O_SYNC` werden die Informationen zur belegten Festplattenkapazität nicht unbedingt sofort aktualisiert.

Bei einem UFS-Dateisystem protokolliert der `du`-Befehl die Größe des Datenblocks innerhalb der Datei. Bei einem ZFS-Dateisystem protokolliert `du` die tatsächliche Größe der Datei wie auf der Festplatte gespeichert. Die Größe umfasst Metadaten sowie Komprimierung. Mit diesen Berichten kann die Frage "Wie viel Speicherplatz erhalte ich zusätzlich, wenn ich diese Datei entferne?" einfacher beantwortet werden. Auf diese Weise sehen Sie selbst wenn die Komprimierung deaktiviert ist, unterschiedliche Ergebnisse zwischen ZFS und UFS.

Wenn Sie die Speicherplatzbelegung, die von dem Befehl `df` angegeben wird, mit dem Befehl `zfs list` vergleichen, beachten Sie, dass `df` die Poolgröße und nicht nur die Größe der Dateisysteme angibt. Außerdem berücksichtigt `df` untergeordnete Dateisysteme oder das Vorhandensein von Schnappschüssen nicht. Wenn ZFS-Eigenschaften, wie Komprimierung und Quota für Dateisysteme festgelegt sind, ist der Abgleich der Speicherplatzbelegung, die von `df` protokolliert wird, möglicherweise schwierig.

Beachten Sie die folgenden Szenarios, die sich auch auf die protokollierte Speicherplatzbelegung auswirken können:

- Bei Dateien, die größer sind als `recordsize` ist der letzte Block der Datei im Allgemeinen halb voll. Wenn der Standardwert für `recordsize` auf 128 KB festgelegt ist, werden pro Datei ca. 64 KB nicht genutzt, was schwerwiegende Auswirkungen haben kann. Die Integration von RFE 6812608 würde dieses Problem lösen. Sie können dieses Problem durch Aktivierung der Komprimierung umgehen. Selbst wenn die Daten bereits komprimiert sind, wird der nicht belegte Teil des letzten Blockes mit Null gefüllt und kann ebenfalls gut komprimiert werden.
- Bei einem RAIDZ-2-Pool belegt jeder Block mindestens 2 Sektoren (512-Byte-Blöcke) mit Paritätsinformationen. Der von den Paritätsinformationen belegte Speicherplatz wird nicht protokolliert. Da er jedoch variieren und bei kleineren Blöcken einen wesentlich größeren Prozentsatz darstellen kann, kann sich dies auf die Protokollierung des Speicherplatzes auswirken. Die Auswirkung ist wesentlich größer, wenn `recordsize` auf 512 Byte festgelegt ist, wobei jeder logische 512-Byte-Block 1,5 KB belegt (3-mal mehr als der ursprüngliche Speicherplatz). Wenn die Speicherplatzeffizienz ungeachtet der Daten, die gespeichert werden, für Sie wichtig ist, sollten Sie den Standardwert von `recordsize` (128 KB) beibehalten und die Komprimierung (mit dem Standardwert `lzjb`) aktivieren.
- Der Befehl `df` berücksichtigt dedupli zierte Dateidaten nicht.

Verhalten bei ungenügendem Speicherplatz

Schnappschüsse von Dateisystemen sind in ZFS äußerst einfach und ohne hohen Aufwand zu erstellen. Schnappschüsse sind in den meisten ZFS-Umgebungen vorhanden. Weitere Informationen zu ZFS-Schnappschüssen finden Sie in Kapitel 6, „Arbeiten mit Oracle Solaris ZFS-Schnappschüssen und -Klonen“.

Das Vorhandensein von Schnappschüssen kann beim Freigeben von Festplattenkapazität ein unvorgesehenes Verhalten hervorrufen. Normalerweise können Sie mit den entsprechenden Zugriffsrechten eine Datei aus einem vollständigen Dateisystem löschen, wodurch im Dateisystem mehr Festplattenkapazität verfügbar wird. Wenn die zu löschende Datei jedoch in einem Schnappschuss eines Dateisystems vorhanden ist, wird durch das Löschen dieser Datei keine Festplattenkapazität freigegeben. Die von dieser Datei belegten Datenblöcke werden weiterhin vom Schnappschuss referenziert.

Deswegen kann infolge des Löschens einer Datei mehr Festplattenkapazität belegt werden, da eine neue Verzeichnisversion erstellt werden muss, die den neuen Namespace-Status widerspiegelt. Dadurch können beim Löschen von Dateien unvorhergesehene `ENOSPC`- oder `EDQUOT`-Fehler auftreten.

Einhängen von ZFS-Dateisystemen

ZFS soll die Komplexität verringern und die Administration vereinfachen. Bei herkömmlichen Dateisystemen müssen Sie beispielsweise bei jedem Hinzufügen eines neuen Dateisystems die

Datei `/etc/vfstab` entsprechend ändern. In ZFS ist das nicht mehr erforderlich, da Dateisysteme je nach den Dateisystemeigenschaften automatisch ein- und ausgehängt werden. ZFS-Einträge müssen also nicht in der Datei `/etc/vfstab` verwaltet werden.

Weitere Informationen zum Einhängen und Freigeben von ZFS-Dateisystemen finden Sie unter „Einhängen von ZFS-Dateisystemen“ auf Seite 203.

Herkömmliche Datenträgeradministration

Wie bereits unter „Speicherpools in ZFS“ auf Seite 26 erwähnt, ist für ZFS kein spezielles Serviceprogramm zur DatenträgerAdministration erforderlich. ZFS arbeitet mit im raw-Modus betriebenen Geräten. Deswegen können Speicher-Pools erstellt werden, die aus (software- oder hardwarebasierten) Logical Volumes bestehen. Diese Konfiguration wird jedoch nicht empfohlen, da ZFS am besten mit im raw-Modus betriebenen Geräten arbeitet. Die Verwendung von Logical Volumes kann sich negativ auf die Leistung und die Zuverlässigkeit auswirken und sollte deswegen vermieden werden.

Solaris ACL-Modell basierend auf NFSv4

Frühere Versionen des Betriebssystems Solaris unterstützten eine auf der POSIX-Spezifikation beruhende Implementierung von Zugriffskontrolllisten. POSIX-basierte Zugriffskontrolllisten dienen zum Schutz von UFS-Dateien. Zum Schutz von ZFS-Dateien wird ein neues, auf der NFSv4-Spezifikation beruhendes Solaris-Modell für Zugriffskontrolllisten verwendet.

Die Hauptunterschiede des neuen Solaris-Zugriffskontrolllistenmodells bestehen in Folgendem:

- Das Modell basiert auf der NFSv4-Spezifikation und ist den NT-Zugriffskontrolllistenmodellen ähnlich.
- Das Modell bietet feiner abgestimmte Zugriffsrechte.
- Zugriffskontrolllisten werden mit den Befehlen `chmod` und `ls` anstatt `setfacl` und `getfacl` eingestellt und angezeigt.
- Das Modell bietet eine reichhaltigere Vererbungssemantik zum Festlegen der Weitergabe von Zugriffsrechten von über- an untergeordnete Verzeichnisse usw.

Weitere Informationen zu Zugriffskontrolllisten mit ZFS-Dateien finden Sie in Kapitel 7, „Schützen von Oracle Solaris ZFS-Dateien mit Zugriffskontrolllisten und Attributen“.

◆ ◆ ◆ KAPITEL 2

Erste Schritte mit Oracle Solaris ZFS

Dieses Kapitel enthält schrittweise Anleitungen zum Einrichten einer einfache Oracle Solaris ZFS-Konfiguration. Wenn Sie dieses Kapitel vollständig durchgearbeitet haben, sollte Ihnen das Funktionsprinzip von ZFS-Befehlen klar sein, und Sie sollten einfache Pools und Dateisysteme erstellen können. Dieses Kapitel ist nicht als allumfassende Informationsquelle gedacht und enthält Verweise auf die nächsten Kapitel, die ausführlichere Informationen enthalten.

Dieses Kapitel enthält die folgenden Abschnitte:

ZFS-Zugriffsrechtsprofile

Wenn Sie ZFS-Administrationsaufgaben ohne das Superuser-Benutzerkonto (Root) durchführen wollen, können Sie zum Ausführen von ZFS-Administrationsaufgaben mithilfe der folgenden Profile eine Rolle annehmen:

- ZFS-SpeicherplatzAdministration – Berechtigung zum Erstellen, Löschen und Ändern von Datenspeichergeräten in einem ZFS-Speicher-Pool
- ZFS-DateisystemAdministration – Berechtigung zum Erstellen, Löschen und Ändern von ZFS-Dateisystemen

Weitere Informationen zum Erstellen bzw. Annehmen von Rollen finden Sie im *System Administration Guide: Security Services* .

Zusätzlich zur Arbeit mit RBAC-Rollen für die Administration von ZFS-Dateisystemen empfiehlt sich der Einsatz der delegierten ZFS-Administration für verteilte ZFS-Administrationsaufgaben. Weitere Informationen finden Sie in Kapitel 8, „Delegierte Oracle Solaris ZFS-Administration“.

Hardware- und Softwarevoraussetzungen und -Empfehlungen für ZFS

Überprüfen Sie vor dem Arbeiten mit der ZFS-Software die folgenden Anforderungen an und Empfehlungen für Hardware und Software:

- Verwenden Sie ein SPARC oder x86-basiertes System, auf dem eine unterstützte Oracle Solaris-Version läuft.
- Der für einen Speicher-Pool mindestens benötigte Speicherplatz beträgt 64 MB. Die Mindestkapazität für Festplatten beträgt 128 MB.
- Im Hinblick auf gute ZFS-Leistung legen Sie die Größe der Speicheranforderung entsprechend Ihrer Workload fest.
- Für die Erstellung einer Poolkonfiguration mit Datenspiegelung verwenden Sie mehrere Controller.

Erstellen eines einfachen ZFS-Dateisystems

Die ZFS-Administration wurde unter Berücksichtigung größtmöglicher Einfachheit entwickelt. Eines der konzeptionellen Ziele besteht in der Verringerung der Anzahl der Befehle, die zum Erstellen eines funktionierenden Dateisystems erforderlich sind. Beispielsweise wird beim Erstellen eines neuen Pools automatisch ein neues ZFS-Dateisystem erstellt und eingehängt.

Das folgende Beispiel zeigt, wie mit einem einzigen Befehl ein einfacher gespiegelter Speicher-Pool namens `tank` und ein ZFS-Dateisystem mit dem Namen `tank` erstellt werden können. Es wird angenommen, dass die Datenträger `/dev/dsk/c1t0d0` und `/dev/dsk/c2t0d0` vollständig verfügbar sind.

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

Weitere Informationen zu ZFS-Pool-Konfigurationen finden Sie unter „Replikationsfunktionen eines ZFS-Speicher-Pools“ auf Seite 49.

Das neue ZFS-Dateisystem `tank` kann so viel Festplattenkapazität wie erforderlich belegen und wird automatisch unter `/tank` eingehängt.

```
# mkfile 100m /tank/foo
# df -h /tank
```

```
Filesystem             size   used  avail capacity  Mounted on
tank                   80G   100M    80G     1%    /tank
```

Innerhalb eines Pools werden Sie wahrscheinlich weitere Dateisysteme erstellen. Dateisysteme bieten Administrationsschnittstellen, mit deren Hilfe verschiedene Datengruppen innerhalb des gleichen Pools verwaltet werden können.

Das folgende Beispiel zeigt, wie ein Dateisystem namens `fs` im Speicher-Pool `tank` erstellt wird.

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

Das neue ZFS-Dateisystem `tank/fs` kann so viel Festplattenkapazität wie erforderlich belegen und wird automatisch unter `/tank/fs` eingehängt.

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

Sie werden wahrscheinlich eine Dateisystemhierarchie erstellen, die auf die Organisationsstruktur Ihrer Einrichtung abgestimmt ist. Weitere Informationen zum Erstellen einer Hierarchie von ZFS-Dateisystemen finden Sie unter „Erstellen einer ZFS-Dateisystemhierarchie“ auf Seite 41.

Erstellen eines einfachen ZFS-Speicherpools

Das vorherige Beispiel verdeutlicht die Einfachheit von ZFS. Im Folgenden wird ein umfassenderes Beispiel vorgestellt, das einer praktischen Anwendung schon relativ nahe kommt. Zuerst müssen Sie Speicheranforderungen definieren und einen Speicher-Pool erstellen. Der Pool beschreibt die physischen Eigenschaften des Speicherbedarfs und muss vor dem Erstellen von Dateisystemen angelegt werden.

▼ So definieren Sie Speicheranforderungen für einen ZFS-Speicher-Pool

1 Legen Sie verfügbare Datenspeichergeräte für den Speicher-Pool fest.

Vor dem Erstellen eines Speicher-Pools müssen Sie festlegen, auf welchen Datenspeichergeräten Daten gespeichert werden sollen. Solche Datenspeichergeräte müssen eine Kapazität von mindestens 128 MB besitzen und dürfen nicht von anderen Bereichen des Betriebssystems verwendet werden. Die Datenspeichergeräte können einzelne Bereiche eines vorformatierten Datenträgers oder gesamte Datenträger sein, die ZFS als einzelnen größeren Bereich formatiert.

Im Speicherbeispiel unter „So erstellen Sie einen ZFS-Speicher-Pool“ auf Seite 40 wird angenommen, dass die gesamten Datenträger `/dev/dsk/c1t0d0` und `/dev/dsk/c2t0d0` verfügbar sind.

Weitere Informationen zu Datenträgern und ihrer Verwendung und Bezeichnung finden Sie unter „Verwenden von Datenträgern in einem ZFS-Speicher-Pool“ auf Seite 45.

2 Wählen Sie die Art der Datenreplikation.

ZFS unterstützt mehrere Datenreplikationsverfahren. Diese bestimmen, gegen welche Hardware-Ausfälle ein Pool gewappnet ist. ZFS unterstützt nicht redundante Konfigurationen (Stripes) sowie Datenspiegelung und RAID-Z (eine RAID-5-Variante).

Im Speicherbeispiel unter „So erstellen Sie einen ZFS-Speicher-Pool“ auf Seite 40 wird eine einfache Datenspiegelung zweier verfügbarer Festplatten verwendet.

Weitere Informationen zur ZFS-Datenreplikation finden Sie unter „Replikationsfunktionen eines ZFS-Speicher-Pools“ auf Seite 49.

▼ So erstellen Sie einen ZFS-Speicher-Pool

1 Melden Sie sich als Root oder in einer gleichberechtigten Rolle (mithilfe des entsprechenden Zugriffsrechtssprofils von ZFS) an.

Weitere Informationen zu ZFS-Zugriffsrechtsprofilen finden Sie unter „ZFS-Zugriffsrechtsprofile“ auf Seite 37.

2 Legen Sie einen Namen für den Speicher-Pool fest.

Dieser Name dient zur Identifizierung des Speicher-Pools bei Verwendung der Befehle `zpool` und `zfs`. Wählen Sie einen Poolnamen Ihrer Wahl, er muss allerdings den Benennungsanforderungen in „Konventionen für das Benennen von ZFS-Komponenten“ auf Seite 31 entsprechen.

3 Erstellen Sie den Pool.

Sie können beispielsweise mit dem folgenden Befehl einen Pool namens `tank` mit Datenspiegelung erstellen:

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

Wenn Datenspeichergeräte andere Dateisysteme enthalten oder anderweitig belegt sind, kann der Befehl den Pool nicht erstellen.

Weitere Informationen zum Erstellen von Speicherpools finden Sie unter „Erstellen von ZFS-Speicherpools“ auf Seite 52. Weitere Informationen darüber, wie Sie herausfinden können, ob Datenspeichergeräte belegt sind oder nicht, finden Sie unter „Erkennen belegter Geräte“ auf Seite 60.

4 **Überprüfen Sie die Ergebnisse.**

Mit dem Befehl `zpool list` können Sie überprüfen, ob der Pool erfolgreich erstellt wurde.

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

Weitere Informationen zum Anzeigen des Pool-Status finden Sie unter „Abfragen des Status von ZFS-Speicher-Pools“ auf Seite 88.

Erstellen einer ZFS-Dateisystemhierarchie

Nach dem Erstellen eines Speicher-Pools zum Speichern von Daten können Sie mit dem Anlegen der Dateisystemhierarchie beginnen. Mit Hierarchien können Informationen auf einfache und gleichzeitig leistungsfähige Weise organisiert werden. Jedem, der bereits mit einem Dateisystem gearbeitet hat, sind diese vertraut.

In ZFS können Dateisysteme in Hierarchien organisiert werden, wobei jedes Dateisystem nur ein übergeordnetes System besitzt. Der Pool-Name ist stets der Name der obersten Hierarchieebene. ZFS erweitert diese Hierarchie durch Eigenschaftsvererbung, sodass gemeinsame Eigenschaften schnell und einfach an gesamten Dateisystemhierarchien gesetzt werden können.

▼ So legen Sie eine ZFS-Dateisystemhierarchie fest

1 **Wählen Sie die Dateisystemgranularität aus.**

ZFS-Dateisysteme spielen für die Administration eine zentrale Rolle. Sie sind kompakt und können schnell erstellt werden. Eine gute Faustregel ist die Erstellung eines Dateisystems pro Benutzer bzw. Projekt, da ein solches Modell die benutzer- bzw. projektweise Kontrolle von Eigenschaften, Schnappschüsse und Sicherungskopien ermöglicht.

Im Beispiel unter „So erstellen Sie ZFS-Dateisysteme“ auf Seite 42 werden zwei ZFS-Dateisysteme (`jeff` und `bill`) erstellt.

Weitere Informationen zur Administration von Dateisystemen finden Sie in Kapitel 5, „Verwalten von Oracle Solaris ZFS-Dateisystemen“.

2 **Fassen Sie ähnliche Dateisysteme zu Gruppen zusammen.**

In ZFS können Dateisysteme in Hierarchien organisiert werden, was die Gruppierung ähnlicher Dateisysteme ermöglicht. Dieses Modell bietet eine zentrale Administrationsschnittstelle zur Kontrolle von Eigenschaften und Administration von Dateisystemen. Ähnliche Dateisysteme sollten unter einem gemeinsamen Namen erstellt werden.

Im Beispiel unter „So erstellen Sie ZFS-Dateisysteme“ auf Seite 42 befinden sich die beiden Dateisysteme in einem Dateisystem namens `home`.

3 Wählen Sie die Dateisystemeigenschaften.

Die meisten Charakteristika von Dateisystemen werden mit Eigenschaften festgelegt. Diese Eigenschaften steuern verschiedene Verhaltensaspekte, so z. B. wo Dateisysteme eingehängt werden, ob und wie sie über das Netzwerk zugänglich sind, ob sie Datenkomprimierung verwenden und ob Kontingente gelten.

Im Beispiel unter „So erstellen Sie ZFS-Dateisysteme“ auf Seite 42 sind alle home-Verzeichnisse unter `/export/zfs/` *user* eingehängt, mithilfe von NFS über das Netzwerk zugänglich und nutzen Datenkomprimierung. Außerdem wird für den Benutzer `jeff` ein Kontingent von 10 GB erzwungen.

Weitere Informationen zu Eigenschaften finden Sie unter „ZFS-Eigenschaften“ auf Seite 181.

▼ So erstellen Sie ZFS-Dateisysteme

1 Melden Sie sich als Root oder in einer gleichberechtigten Rolle (mithilfe des entsprechenden Zugriffsrechtssprofils von ZFS) an.

Weitere Informationen zu ZFS-Zugriffsrechtsprofilen finden Sie unter „ZFS-Zugriffsrechtsprofile“ auf Seite 37.

2 Erstellen Sie die gewünschte Hierarchie.

In diesem Beispiel wird ein Dateisystem erstellt, das als Container für untergeordnete Dateisysteme dienen soll.

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

3 Legen Sie die vererbten Eigenschaften fest.

Nach dem Erstellen der Dateisystemhierarchie sollten Eigenschaften festgelegt werden, die für alle Benutzer gleich sind:

```
# 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
```

Sie können Dateisystemeigenschaften beim Erstellen des Dateisystems festlegen. Beispiel:

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

Weitere Informationen zu Eigenschaften und deren Vererbung finden Sie unter „ZFS-Eigenschaften“ auf Seite 181.

Als Nächstes werden einzelne Dateisysteme im Dateisystem `home` des Pools `tank` gruppiert.

4 Erstellen Sie die einzelnen untergeordneten Dateisysteme.

Wären die Dateisysteme bereits erstellt, könnten die Eigenschaften anschließend auf der `home`-Ebene geändert werden. Alle Eigenschaften können dynamisch geändert werden, wenn Dateisysteme in Betrieb sind.

```
# zfs create tank/home/jeff
# zfs create tank/home/bill
```

Diese Dateisysteme erben ihre Eigenschaftswerte von ihrem übergeordneten Dateisystem. Deswegen werden sie automatisch unter `/export/zfs/`*user* eingehängt und sind mit NFS über das Netzwerk zugänglich. Sie brauchen die Datei `/etc/vfstab` bzw. `/etc/dfs/dfstab` nicht zu bearbeiten.

Weitere Informationen zum Erstellen von Dateisystemen finden Sie unter „Erstellen eines ZFS-Dateisystems“ auf Seite 178.

Weitere Informationen zum Einhängen und Freigeben von Dateisystemen finden Sie unter „Einhängen von ZFS-Dateisystemen“ auf Seite 203.

5 Legen Sie die dateisystemspezifischen Eigenschaften fest.

In diesem Beispiel ist dem Benutzer `jeff` ein Kontingent von 10 GB zugewiesen. Diese Eigenschaft beschränkt unabhängig davon, wieviel Speicherplatz im Pool verfügbar ist, den durch diesen Benutzer zu belegenden Speicherplatz.

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

6 Überprüfen Sie die Ergebnisse.

Mit dem Befehl `zfs list` können verfügbare Dateisysteminformationen angezeigt werden:

```
# 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/bill          8K  67.0G     8K  /export/zfs/bill
tank/home/jeff          8K  10.0G     8K  /export/zfs/jeff
```

Beachten Sie, dass der Benutzer `jeff` maximal 10 GB Speicherplatz belegen darf, der Benutzer `bill` dagegen die gesamte Pool-Kapazität (67 GB) nutzen kann.

Weitere Informationen zum Anzeigen des Dateisystemstatus finden Sie unter „Abfragen von ZFS-Dateisysteminformationen“ auf Seite 195.

Weitere Informationen zur Belegung und Berechnung von Festplattenkapazität finden Sie unter „Berechnung von ZFS-Festplattenkapazität“ auf Seite 32.

◆ ◆ ◆ KAPITEL 3

Verwalten von Oracle Solaris ZFS-Speicher-Pools

In diesem Kapitel wird beschrieben, wie Speicher-Pools in Oracle Solaris ZFS erstellt und verwaltet werden können.

Dieses Kapitel enthält die folgenden Abschnitte:

Komponenten eines ZFS-Speicher-Pools

Die folgenden Abschnitte enthalten ausführliche Informationen zu den folgenden Komponenten eines Speicher-Pools:

Verwenden von Datenträgern in einem ZFS-Speicher-Pool

Das grundlegendste Element eines Speicher-Pools ist physischer Speicher. Bei physischem Speicher kann es sich um eine Blockeinheit mit mindestens 128 MB Speicherkapazität handeln. Normalerweise handelt es sich dabei um Festplatten, die vom System im Verzeichnis `/dev/dsk` erkannt werden.

Ein Datenspeichergerät kann eine gesamte Festplatte (z. B. `c1t0d0`) oder ein einzelner Festplattenbereich (z. B. `c0t0d0s7`) sein. Es wird empfohlen, eine gesamte Festplatte zu verwenden, da die Festplatte in diesem Fall nicht formatiert werden muss. ZFS formatiert solche Datenträger mit einem EFI-Label als einzelnen, großen Bereich. In diesem Fall wird die vom Befehl `format` ausgegebene Partitionstabelle in etwa wie folgt angezeigt:

```
Current partition table (original):
Total disk sectors available: 143358287 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector         Size         Last Sector
  0        usr    wm               256       68.36GB          143358320
  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         143358321        8.00MB          143374704
```

Beachten Sie folgende Punkte bei der Verwendung von ganzen Festplatten in Ihren ZFS-Speicherpools:

- Bei Verwendung einer ganzen Festplatte muss diese der Benennungskonvention `/dev/dsk/cNtNdN` entsprechen. Einige Treiber von Drittanbietern nutzen andere Benennungskonventionen oder hängen Datenträger nicht im Verzeichnis `/dev/dsk`, sondern anderswo ein. Damit solche Datenträger verwendet werden können, müssen Sie diese manuell benennen und für ZFS einen Bereich verfügbar machen.
- Auf einem x86-System muss die Festplatte eine gültige Solaris-`fdisk`-Partition enthalten. Weitere Informationen zum Erstellen oder Ändern einer Solaris `fdisk`-Partition finden Sie unter „Setting Up Disks for ZFS File Systems (Task Map)“ in *System Administration Guide: Devices and File Systems*.
- ZFS verwendet EFI-Label, wenn Sie Speicher-Pools mit gesamten Festplatten erstellen. Weitere Informationen zu EFI-Labels finden Sie unter „EFI (GPT) Disk Label“ in *System Administration Guide: Devices and File Systems*.

Datenträger können mit dem vollständigen Pfad (z. B. `/dev/dsk/c1t0d0` oder in einer Kurzschreibweise, die aus dem Gerätenamen im Verzeichnis `/dev/dsk` besteht (z. B. `c1t0d0`), angegeben werden. Im Folgenden sind beispielsweise gültige Datenträgernamen aufgeführt:

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

Verwenden von Bereichen in einem ZFS-Speicher-Pool

Festplatten können mit einem Solaris VTOC-(SMI-)Label bezeichnet werden, wenn Sie einen Speicherpool mit einem Festplattenbereich erstellen. Die Verwendung von Festplattenbereichen für einen Pool wird jedoch nicht empfohlen, weil die Verwaltung von Festplattenbereichen schwieriger ist.

In einem SPARC-System verfügt eine 72-GB-Festplatte über 68 GB nutzbaren Speicher im Bereich 0, wie in der folgenden Ausgabe von `format` zu sehen ist:

```
# 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
```

In einem x86-System verfügt eine 72-GB-Festplatte über 68 GB nutzbaren Speicher im Bereich 0, wie in der folgenden Ausgabe von `format` zu sehen ist. Einige Boot-Infomationen sind in Bereich 8 enthalten. Bereich 8 erfordert keine Administration und kann nicht verändert werden.

```
# 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
```

Eine `fdisk`-Partition ist auch auf einem x86-basierten System vorhanden. Eine `fdisk`-Partition wird durch einen `/dev/dsk/cN[tN]dNpN`-Gerätenamen dargestellt und dient als Container für die verfügbaren Festplattenbereiche. Verwenden Sie kein `cN[tN]dNpN`-Gerät für eine Komponente eines ZFS-Speicher-Pools, da diese Konfiguration weder getestet wurde noch unterstützt wird.

Verwenden von Dateien in einem ZFS-Speicher-Pool

In ZFS können Sie Dateien als virtuelle Geräte im Speicherpool verwenden. Dieses Merkmal dient vorrangig zum Testen und Aktivieren einfacher experimenteller Konfigurationen und ist nicht für den Praxiseinsatz unter Produktionsbedingungen gedacht.

- Wenn Sie ein ZFS-Pool mit Dateien aus einem UFS-Dateisystem erstellen, verlassen sie sich stillschweigend darauf, dass UFS ordnungsgemäße und synchrone Semantik gewährleistet.
- Wenn Sie einen ZFS-Pool erstellen, der von Dateien oder Volumes gesichert wird, die in einem anderen ZFS-Pool erstellt wurden, kann es zu einer Sperre oder einem Absturz des Systems kommen.

Dateien können jedoch nützlich sein, wenn Sie ZFS zum ersten Mal testen oder mit komplexeren Konfigurationen experimentieren und nicht genügend physische Datenspeichergeräte zur Verfügung stehen. Alle Dateien müssen mit dem vollständigen Pfad angegeben werden und mindestens 64 MB groß sein.

Überlegungen zu ZFS-Speicherpools

Beachten Sie Folgendes, wenn Sie ZFS-Speicherpools erstellen und verwalten.

- Die Verwendung gesamter physischer Festplatten ist die einfachste Methode zum Erstellen von ZFS-Speicher-Pools. Was die Administration, Zuverlässigkeit und Leistung angeht, so werden ZFS-Konfigurationen weitaus komplexer, wenn Sie Pools aus Datenträgerbereichen, LU-Nummern in Hardware-RAID-Arrays oder Volumes, die von Datenträgeradministrationssoftware bereitgestellt werden, aufbauen. Bei der Entscheidung, ob Sie ZFS mit anderen hardware- bzw. softwarebasierten Speicherlösungen konfigurieren, sollten Sie die folgenden Aspekte berücksichtigen:
  - Wenn Sie eine ZFS-Konfiguration mit LU-Nummern aus Hardware-RAID-Arrays aufbauen, muss Ihnen die Beziehung zwischen den Redundanzfunktionen von ZFS und denen des Arrays klar sein. Bestimmte Konfigurationen bieten ausreichende Redundanz und Leistung, bei anderen braucht dies aber nicht der Fall zu sein.
  - Sie können logische Geräte für ZFS mithilfe von Volumes aufbauen, die von Software zur DatenträgerAdministration wie z. B. Solaris Volume Manager (SVM) oder Veritas Volume Manager (VxVM) bereitgestellt werden. Solche Konfigurationen werden jedoch nicht empfohlen. Obwohl ZFS mit solchen Konfigurationen ordnungsgemäß funktioniert, ist die Leistung in diesen Fällen oftmals nicht optimal.

Weitere Informationen zu Empfehlungen für Speicherpools finden Sie in Kapitel 11, „Empfohlene Oracle Solaris ZFS-Vorgehensweisen“.

- Datenträger werden nach Pfad und Geräte-ID (falls verfügbar) identifiziert. Bei Systemen, in denen Geräte-ID-Informationen zur Verfügung stehen, erlaubt diese Identifizierungsmethode, Geräte neu zu konfigurieren, ohne dass eine Aktualisierung von ZFS nötig ist. Die Geräte-ID-Generierung und -Administration kann von System zu System unterschiedlich sein. Darum sollten Sie den Pool zunächst exportieren und erst dann Geräte verschieben (beispielsweise eine Festplatte von einem Controller zu einem anderen). Ein Systemereignis wie beispielsweise eine Firmware-Aktualisierung oder eine andere Hardware-Änderung kann die Geräte-IDs im ZFS-Speicher-Pool verändern. Dadurch kann bewirkt werden, dass die Geräte nicht mehr zur Verfügung stehen.

Replikationsfunktionen eines ZFS-Speicher-Pools

ZFS bietet in Konfigurationen mit Datenspiegelung und RAID-Z Datenredundanz und Selbstheilungsfunktionen.

Speicher-Pools mit Datenspiegelung

Für Speicher-Pools mit Datenspiegelung sind mindestens zwei Datenträger erforderlich, auf die optimalerweise von zwei verschiedenen Controllern zugegriffen werden sollte. Für eine Datenspiegelung können viele Datenträger verwendet werden. Darüber hinaus können Sie in einem Pool mehrere Datenspiegelungen erstellen. Eine einfache Datenspiegelungskonfiguration würde im Prinzip ungefähr wie folgt aussehen:

```
mirror c1t0d0 c2t0d0
```

Eine komplexere Datenspiegelungskonfiguration würde im Prinzip ungefähr wie folgt aussehen:

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

Informationen zum Erstellen von Speicher-Pools mit Datenspiegelung finden Sie unter „Erstellen eines Speicher-Pools mit Datenspiegelung“ auf Seite 53.

Speicher-Pools mit RAID-Z-Konfiguration

Neben Speicher-Pools mit Datenspiegelung bietet ZFS RAID-Z-Konfiguration, die Fehlertoleranzen mit ein-, zwei- oder dreifacher Parität aufweisen. RAID-Z mit einfacher Parität (`raidz` oder `raidz1`) ist mit RAID-5 vergleichbar. RAID-Z mit doppelter Parität (`raidz2`) ähnelt RAID-6.

Weitere Informationen über RAIDZ-3 (`raidz3`) finden Sie im folgenden Blog:

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

Alle herkömmlichen Algorithmen, die RAID-5 ähnlich sind (z.B. RAID-4, RAID-5, RAID-6, RDP und EVEN-ODD), können von einem Problem betroffen sein, das als so genanntes *RAID-5 Write Hole* bekannt ist. Wenn nur ein Teil eines RAID-5-Bereichs geschrieben wird und ein Stromausfall auftritt, bevor alle Datenblöcke auf den Datenträger geschrieben wurden, kann die Parität der Daten nicht hergestellt werden, wodurch diese Daten nutzlos sind (es sei denn, sie werden von einem nachfolgenden vollständigen Bereich überschrieben). Bei RAID-Z verwendet ZFS RAID-Stripes veränderlicher Länge, sodass alle Schreibvorgänge vollständige Stripes speichern. Dieses Design ist nur möglich, weil ZFS die Administration von Dateisystemen und Datenspeichergeräten so integriert, dass die Metadaten eines Dateisystems genügend Informationen zum zugrunde liegenden Datenredundanzmodell besitzen und sie somit RAID-Stripes veränderlicher Länge verarbeiten können. RAID-Z ist die weltweit erste nur auf Software basierte Lösung, die das RAID-5 Write Hole eliminiert.

Eine RAID-Z-Konfiguration mit N Datenträgern der Kapazität X mit P Paritätsdatenträgern kann ca. (N-P)*X Byte speichern und ist gegen einen Ausfall von P Geräten ohne Gefährdung der Datenintegrität geschützt. Für eine RAID-Z-Konfiguration mit einfacher Parität sind mindestens zwei, für eine RAID-Z-Konfiguration mit doppelter Parität mindestens drei Datenträger erforderlich usw. Wenn sich beispielsweise in einer RAID-Z-Konfiguration mit einfacher Parität drei Festplatten befinden, belegen Paritätsdaten eine Festplattenkapazität, die der Kapazität einer der drei Festplatten entspricht. Zum Erstellen einer RAID-Z-Konfiguration ist darüber hinaus keine spezielle Hardware erforderlich.

Eine RAID-Z-Konfiguration mit drei Datenträgern würde im Prinzip ungefähr wie folgt aussehen:

```
raidz c1t0d0 c2t0d0 c3t0d0
```

Eine komplexere RID-Z-Konfiguration würde im Prinzip ungefähr wie folgt aussehen:

```
raidz c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0 c6t0d0 c7t0d0
raidz c8t0d0 c9t0d0 c10t0d0 c11t0d0c12t0d0 c13t0d0 c14t0d0
```

Wenn Sie RAID-Z-Konfiguration mit mehreren Datenträgern erstellen, sollten die Datenträger in mehrere Gruppen aufgeteilt werden. Beispielsweise sollten Sie eine RAID-Z-Konfiguration mit 14 Datenträgern besser in zwei Gruppen mit je 7 Datenträgern aufteilen. RAID-Z-Konfigurationen mit weniger als zehn Datenträgern erreichen eine optimalere Leistung.

Informationen zum Erstellen eines RAID-Z-Speicher-Pools finden Sie unter „Erstellen eines RAID-Z-Speicher-Pools“ auf Seite 54.

Weitere Informationen zur Auswahl zwischen einer Datenspiegelungs- oder RAID-Z-Konfiguration, die auf Leistungs- und Festplattenkapazitätskriterien basiert, finden Sie im folgenden Blog:

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

Weitere Informationen zu Empfehlungen für RAID-Z-Speicherpools finden Sie in Kapitel 11, „Empfohlene Oracle Solaris ZFS-Vorgehensweisen“.

ZFS-Hybrid-Speicher-Pool

Der ZFS-Hybrid-Speicher-Pool, der zur Sun Storage 7000-Produktreihe von Oracle gehört, ist ein spezieller Speicher-Pool, der DRAM, SSDs und HDDs zur Erhöhung von Leistung und Kapazität bei gleichzeitiger Verringerung des Energieverbrauchs kombiniert. Sie können die ZFS-Redundanzkonfiguration des Speicher-Pools auswählen und einfach andere Konfigurationsoptionen mit dieser Administrationsschnittstelle des Produkts verwalten.

Weitere Informationen zu diesem Produkt finden Sie im *Sun Storage Unified Storage System Administration Guide* .

Selbstheilende Daten in einer redundanten Konfiguration

ZFS bietet in Konfigurationen mit Datenspiegelung und RAID-Z Selbstheilungsfunktionen.

Bei Erkennung beschädigter Datenblöcke ruft ZFS nicht nur die unbeschädigten Daten von einer redundanten Kopie ab, sondern repariert die beschädigten Daten auch, indem es diese mit der unbeschädigten Kopie ersetzt.

Dynamisches Striping in einem Speicher-Pool

ZFS verteilt Daten dynamisch mithilfe des Striping-Verfahrens über alle in der obersten Hierarchieebene befindlichen virtuellen Geräte. Die Entscheidung, wo Daten gespeichert werden, wird zur Zeit des Speichervorgangs getroffen.

Wenn neue virtuelle Geräte zu einem Pool hinzugefügt werden, weist ZFS dem neuen Gerät schrittweise Daten zu, wodurch Leistungsrichtlinien und Richtlinien für die Zuweisung von Festplattenkapazität eingehalten werden. Jedes virtuelle Gerät kann auch für die Datenspiegelung oder als RAID-Z-Gerät, das andere Datenträgergeräte bzw. Dateien enthält,

eingesetzt werden. Mit dieser Konfiguration wird die Kontrolle der Fehlercharaketristika eines Pools flexibler. So können Sie beispielsweise aus vier Datenträgern die folgenden Konfigurationen bilden:

- vier Datenträger mit dynamischem Striping
- eine vierfache RAID-Z-Konfiguration
- zwei zweifache Geräte für die Datenspiegelung, die dynamisches Striping verwenden

Obwohl ZFS die Kombination verschiedener Arten virtueller Geräte im gleichen Pool unterstützt, wird dies nicht empfohlen. Sie können beispielsweise einen Pool mit einer zweifachen Datenspiegelungs- und einer dreifachen RAID-Z-Konfiguration erstellen. Die Fehlertoleranz des Gesamtsystems ist jedoch nur so gut wie die die Geräts mit der schlechtesten Fehlertoleranz, in diesem Fall RAID-Z. Eine bewährte Verfahrensweise ist, virtuelle Geräte der obersten Hierarchieebene zu verwenden, die das gleiche Redundanzniveau besitzen.

Erstellen und Entfernen von ZFS-Speicher-Pools

In den folgenden Abschnitten werden unterschiedliche Situationen zum Erstellen und Entfernen von ZFS-Speicher-Pools beschrieben:

- „Erstellen von ZFS-Speicherpools“ auf Seite 52
- „Anzeigen von Informationen zu virtuellen Geräten in Storage-Pools“ auf Seite 59
- „Behandlung von Fehlern beim Erstellen von ZFS-Speicher-Pools“ auf Seite 60
- „Löschen von ZFS-Speicher-Pools“ auf Seite 63

Pools können schnell und einfach erstellt und entfernt werden. Allerdings sollten Sie solche Vorgänge äußerste Vorsicht durchführen. Obwohl Überprüfungen durchgeführt werden, die die Verwendung von Geräten verhindern, die bereits von einem anderen Pool benutzt werden, ist es ZFS nicht immer bekannt, ob ein Datenspeichergerät bereits belegt ist oder nicht. Das Entfernen eines Pools ist einfacher als die Erstellung eines Pools. Benutzen Sie den Befehl `zpool destroy` mit Vorsicht. Die Ausführung dieses einfachen Befehls kann bedeutende Konsequenzen nach sich ziehen.

Erstellen von ZFS-Speicherpools

Speicher-Pools können mit dem Befehl `zpool create` erstellt werden. Dieser Befehl akzeptiert einen Pool-Namen und eine beliebige Anzahl virtueller Geräte als Argumente. Der Pool-Name muss den unter „Konventionen für das Benennen von ZFS-Komponenten“ auf Seite 31 aufgeführten Namenskonventionen genügen.

Erstellen eines einfachen Speicher-Pools

Mit dem folgenden Befehl wird ein Pool namens `tank` erstellt, das aus den beiden Datenträgern `c1t0d0` und `c1t1d0` besteht:

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

Gerätenamen, die ganze Datenträger repräsentieren, befinden sich im Verzeichnis `/dev/dsk` und werden von ZFS als einzelner, großer Bereich gekennzeichnet. Daten werden mithilfe von Striping dynamisch über beide Datenträger verteilt.

Erstellen eines Speicher-Pools mit Datenspiegelung

Verwenden Sie zum Erstellen eines Pools mit Datenspiegelung das Schlüsselwort `mirror`, gefolgt von einer bestimmten Anzahl von Datenspeichergeräten, aus denen das Datenspiegelungssystem bestehen soll Mehrere Datenspiegelungssysteme können durch Wiederholen des Schlüsselworts `mirror` in der Befehlszeile erstellt werden. Mit dem folgenden Befehl wird ein Pool mit zwei zweifachen Datenspiegelungen erstellt:

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

Das zweite Schlüsselwort `mirror` spezifiziert ein neues virtuelles Gerät der obersten Hierarchieebene. Daten werden mithilfe von Striping dynamisch über beide Datenträger verteilt, wobei zwischen jedem Datenträger eine jeweilige Datenredundanz auftritt.

Weitere Informationen zu empfohlenen gespiegelten Konfigurationen finden Sie in Kapitel 11, „Empfohlene Oracle Solaris ZFS-Vorgehensweisen“.

Gegenwärtig werden für ZFS-Datenspiegelungskonfigurationen folgende Vorgänge unterstützt:

- Hinzufügen eines weiteren Festplattensatzes zu einer vorhandenen Datenspiegelungskonfiguration, um eine neues virtuelles Gerät der obersten Hierarchieebene bereitzustellen. Weitere Informationen dazu finden Sie unter „Hinzufügen von Datenspeichergeräten zu einem Speicher-Pool“ auf Seite 65.
- Hinzufügen weiterer Festplatten zu einer vorhandenen Datenspiegelungskonfiguration bzw. Hinzufügen weiterer Festplatten zu einer nicht replizierten Konfiguration, um eine Datenspiegelungskonfiguration zu erstellen. Weitere Informationen dazu finden Sie unter „Verbinden und Trennen von Geräten in einem Speicher-Pool“ auf Seite 69.
- Ersetzen einer oder mehrerer Festplatten in einer vorhandenen Datenspiegelungskonfiguration, solange die Kapazität der neuen Festplatten größer oder gleich der Kapazität der zu ersetzenden Festplatten ist. Weitere Informationen dazu finden Sie unter „Austauschen von Geräten in einem Speicher-Pool“ auf Seite 77.
- Abtrennen einer oder mehrerer Festplatten in einer Datenspiegelungskonfiguration, solange die verbleibenden Speichergeräte für die Konfiguration noch eine ausreichende Redundanz gewährleisten. Weitere Informationen dazu finden Sie unter „Verbinden und Trennen von Geräten in einem Speicher-Pool“ auf Seite 69.
- Teilen einer gespiegelten Konfiguration, indem eine der Festplatten entfernt wird, um einen neuen, identischen Pool zu erstellen. Weitere Informationen finden Sie unter „Erstellen eines neuen Pools durch Teilen eines ZFS-Speicher-Pools mit Datenspiegelung“ auf Seite 71.

Ein Speichergerät, das kein Spare-, Protokollier- oder Cache-Gerät ist, kann nicht direkt aus einem Speicherpool mit Datenspiegelung entfernt werden.

Erstellen eines ZFS-Root-Pools

Beachten Sie die folgenden Voraussetzungen für die Konfiguration von Root-Pools:

- Der Root-Pool muss als gespiegelte Konfiguration oder als einzelne Festplattenkonfiguration erstellt werden. Sie können mit dem Befehl `zpooladd` keine zusätzlichen Festplatten hinzufügen, um mehrere gespiegelte virtuelle Geräte der obersten Hierarchieebene zu erstellen, aber Sie können ein gespiegeltes virtuelles Gerät mit dem Befehl `zpool attach` erweitern.
- RAID-Z- oder Stripes-Konfigurationen werden nicht unterstützt.
- Der Root-Pool kann über kein separates Protokolliergerät verfügen.
- Bei dem Versuch, eine nicht unterstützte Pool-Konfiguration für einen Root-Pool zu verwenden, wird eine Meldung wie die folgende angezeigt:

```
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
```

Weitere Informationen zum Installieren und Booten eines ZFS-Dateisystems finden Sie in Kapitel 4, „Installieren und Booten eines Oracle Solaris ZFS-Root-Dateisystems“.

Erstellen eines RAID-Z-Speicher-Pools

Das Erstellen eines RAID-Z-Pools mit einfacher Parität läuft ähnlich ab wie das Erstellen eines Pools mit Datenspiegelung. Der einzige Unterschied besteht darin, dass anstatt des Schlüsselworts `mirror` das Schlüsselwort `raidz` bzw. `raidz1` verwendet wird. Das folgende Beispiel zeigt die Erstellung eines Pools mit einem RAID-Z-Gerät, das aus fünf Datenträgern besteht:

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

Dieses Beispiel zeigt, dass Datenträger angegeben werden können, indem ihre abgekürzten oder ihre vollständigen Gerätenamen verwendet werden. Sowohl `/dev/dsk/c5t0d0` als auch `c5t0d0` beziehen sich auf denselben Datenträger.

Eine RAID-Z-Konfiguration mit zwei- oder dreifacher Parität kann beim Erstellen eines Pools mithilfe des Schlüsselworts `raidz2` oder `raidsz3` angelegt werden. Beispiel :

```
# 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 c8t0d0
# 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
            c8t0d0  ONLINE       0     0     0
errors: No known data errors
```

Folgende Vorgänge werden in einer ZFS-RAID-Z-Konfiguration gegenwärtig unterstützt:

- Hinzufügen eines weiteren Festplattensatzes zu einer vorhandenen RAID-Z-Konfiguration, um eine neues virtuelles Gerät der obersten Hierarchieebene bereitzustellen. Weitere Informationen dazu finden Sie unter „Hinzufügen von Datenspeichergeräten zu einem Speicher-Pool“ auf Seite 65.
- Ersetzen einer oder mehrerer Festplatten in einer vorhandenen RAID-Z-Konfiguration, solange die Kapazität der neuen Festplatten größer oder gleich der Kapazität der zu ersetzenden Festplatten ist. Weitere Informationen dazu finden Sie unter „Austauschen von Geräten in einem Speicher-Pool“ auf Seite 77.

Folgende Vorgänge werden in einer RAID-Z-Konfiguration gegenwärtig *nicht* unterstützt:

- Hinzufügen einer weiteren Festplatte zu einer vorhandenen RAID-Z-Konfiguration.
- Entfernen einer Festplatte aus einer RAID-Z-Konfiguration, ausgenommen Entfernen einer Festplatte, die durch eine Ersatzfestplatte ersetzt wird, oder Entfernen einer Ersatzfestplatte.
- Ein Speichergerät, das kein Protokollier- oder Cache-Gerät ist, kann nicht direkt aus einer RAID-Z-Konfiguration entfernt werden. Dafür wurde ein RFE eingereicht.

Weitere Informationen zu RAID-Z-Konfigurationen finden Sie unter „Speicher-Pools mit RAID-Z-Konfiguration“ auf Seite 50.

Erstellen eines ZFS-Speicher-Pools mit Protokolliergeräten

Das ZFS-Intent-Log (ZIL) erfüllt die POSIX-Anforderungen für synchrone Transaktionen. So setzen Datenbanken bei der Rückkehr von Systemaufrufen beispielsweise oft voraus, dass Transaktionen auf stabilen Speichergeräten stattfinden. NFS und andere Anwendungen können auch `fsync()` verwenden, um die Datenstabilität zu gewährleisten.

Standardmäßig wird das ZIL aus Blöcken innerhalb des Haupt-Pools zugewiesen. Eine bessere Performance kann jedoch durch Verwendung separater Intent-Log-Geräte erzielt werden, wie NVRAM oder einer dedizierten Festplatte.

Berücksichtigen Sie folgende Aspekte bei der Überlegung, ob die Einrichtung eines ZFS-Protokolliergeräts für Ihre Umgebung ratsam ist:

- Protokolliergeräte für das Intent-Protokoll von ZFS sind etwas Anderes als Datenbankprotokolldateien.
- Jegliche Leistungssteigerung durch die Implementierung eines separaten Protokolliergeräts ist von der Art des Geräts, der Hardwarekonfiguration des Speicher-Pools sowie von der Arbeitslast der Anwendung abhängig. Vorläufige Informationen zur Leistung finden Sie in diesem Blog:

  `http://blogs.oracle.com/perrin/entry/slog_blog_or_blogging_on`
- Es ist möglich, Protokolliergeräte zu spiegeln oder deren Replikation aufzuheben. RAID-Z wird für Protokolliergeräte jedoch nicht unterstützt.
- Wenn ein separates Protokolliergerät nicht gespiegelt wurde und das Gerät mit den Protokollen ausfällt, wird durch Speicherung von Protokollblöcken auf den Speicher-Pool zurückgeschaltet.
- Protokolliergeräte können als Teil des größeren Speicherpools hinzugefügt, ersetzt, entfernt, angehängt, ausgehängt, importiert und exportiert werden.
- Sie können zur Datenspiegelung an ein vorhandenes Protokolliergerät ein weiteres Protokolliergerät anschließen. Dies entspricht dem Verbinden eines Speichergeräts in einem Speicher-Pool ohne Datenspiegelung.
- Die Mindestgröße für ein Protokolliergerät entspricht der Mindestkapazität für jedes Gerät in einem Pool. Dies sind 64 MB. Auf einem Protokolliergerät können verhältnismäßig wenige Ausführungsdaten gespeichert werden. Beim Festschreiben der Protokolltransaktion (Systemaufruf) werden die Protokollblöcke geleert.
- Ein Protokolliergerät sollte nicht größer als rund die Hälfte des physischen Speichers sein, da dies die Höchstmenge an potenziellen Ausführungsdaten ist, die gespeichert werden kann. So sollten Sie beispielsweise für ein System mit 16 GB physischem Speicher ein Protokolliergerät mit 8 GB Speicher in Erwägung ziehen.

Sie können ZFS-Protokolliergeräte während oder nach der Erstellung eines Speicher-Pools einrichten.

Das folgende Beispiel zeigt, wie ein Speicher-Pool mit Datenspiegelung und gespiegelten Protokolliergeräten erstellt wird:

```
# zpool create datap mirror c0t5000C500335F95E3d0 c0t5000C500335F907Fd0 mirror
c0t5000C500335BD117d0 c0t5000C500335DC60Fd0 log mirror c0t5000C500335E106Bd0 c0t5000C500335FC3E7d0
# 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
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0
        logs
          mirror-2                 ONLINE       0     0     0
            c0t5000C500335E106Bd0  ONLINE       0     0     0
            c0t5000C500335FC3E7d0  ONLINE       0     0     0

errors: No known data errors
```

Informationen zum Wiederherstellen des Normalbetriebs nach dem Fehlschlag eines Protokolliergeräts finden Sie in Beispiel 10–2.

Erstellen eines ZFS-Speicher-Pools mit Cache-Geräten

Cache-Speicher bietet zwischen Hauptspeicher und Festplatte eine zusätzliche Schicht zur Datenspeicherung. Cache-Geräte bieten die größte Leistungsverbesserung für die Direktspeicherung von Daten mit vorwiegend statischem Inhalt.

Sie können mit Cache-Geräten einen Pool erstellen, um Speicher-Pool-Daten im Cache zu speichern. Beispiel:

```
# 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
```

Nach dem Erstellen von Cache-Speichergeräten werden diese nach und nach mit Daten aus dem Hauptspeicher gefüllt. Je nach Größe eines definierten Cache-Speichergeräts kann es bis zu über eine Stunde lang dauern, bis das Gerät voll ist. Die Kapazität und Lesevorgänge können mithilfe des Befehls `zpool iostat` wie folgt überwacht werden:

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

Nach der Erstellung eines Pools können Cache-Geräte dem Pool hinzugefügt oder aus dem Pool entfernt werden.

Beachten Sie Folgendes, wenn Sie die Erstellung eines ZFS-Speicher-Pools mit Cache-Geräten in Betracht ziehen:

- Cache-Geräte bieten die größte Leistungsverbesserung für die Direktspeicherung von Daten mit vorwiegend statischem Inhalt.
- Die Kapazität und Lesevorgänge können mithilfe des Befehls `zpool iostat` überwacht werden.
- Beim Erstellen eines Speicher-Pools können eines oder mehrere Cache-Geräte hinzugefügt werden. Die Geräte können auch nach der Erstellung des Pools hinzugefügt oder entfernt werden. Weitere Informationen finden Sie in Beispiel 3–4.
- Cachegeräte können nicht gespiegelt werden oder Teil einer RAID-Z-Konfiguration sein.
- Wenn auf einem Cache-Gerät ein Lesefehler entdeckt wird, wird der betreffende E/A-Lesevorgang wieder an das ursprüngliche Speicher-Pool-Gerät zurückgegeben, das Teil einer gespiegelten oder RAID-Z-Konfiguration sein kann. Der Inhalt der Cache-Geräte wird als flüchtig betrachtet, wie auch bei anderen Cache-Geräten des Systems.

Vorsichtsmaßnahmen beim Erstellen von Speicherpools

Beachten Sie die folgenden Vorsichtsmaßen, wenn Sie ZFS-Speicherpools erstellen und verwalten.

- Festplatten, die Bestandteil eines vorhandenen Speicherpools sind, dürfen nicht neu partitioniert oder neu bezeichnet werden. Wenn Sie versuchen, eine Festplatte eines Root-Pools neu zu partitionieren oder zu bezeichnen, müssen Sie möglicherweise das Betriebssystem neu installieren.
- Erstellen Sie keinen Speicherpool, der Komponenten aus einem anderen Speicherpool enthält, wie Dateien oder Volumes. Bei dieser nicht unterstützten Konfiguration kann es zu Deadlocks kommen.

- Ein Pool, der mit einem einzelnen Bereich oder einer einzelnen Festplatte erstellt wird, hat keine Redundanz. Er kann Datenverlusten ausgesetzt sein. Bei einem Pool, der mit mehreren Bereichen, jedoch ohne Redundanz erstellt wird, kann es ebenfalls zu einem Datenverlust kommen. Ein Pool, der mit mehreren Bereichen über Festplatten hinweg erstellt wird, ist schwieriger zu verwalten als ein Pool, der mit ganzen Festplatten erstellt wird.
- Ein Pool, der nicht mit ZFS-Redundanz (RAIDZ oder Spiegelung) erstellt wird, kann nur Dateninkonsistenzen protokollieren. Er kann keine Dateninkonsistenzen beheben.
- Auch wenn ein Pool, der mit ZFS-Redundanz erstellt wird, Ausfallzeiten wegen Hardwarefehlern verringern kann, ist er nicht gegen Hardwarefehler, Stromausfälle oder getrennte Kabelverbindungen gefeit. Sie müssen Ihre Daten in regelmäßigen Abständen sichern. Routinemäßige Backups von Pooldaten auf unternehmensexterner Hardware sind unerlässlich.
- Ein Pool kann nicht systemübergreifend freigegeben werden. ZFS ist kein Clusterdateisystem.

Anzeigen von Informationen zu virtuellen Geräten in Storage-Pools

Jeder Speicher-Pool enthält eines oder mehrere virtuelle Geräte. Unter einem *virtuellen Gerät* versteht man eine interne Darstellung eines Speicher-Pools zur Beschreibung der Struktur der physischen Datenspeicherung und Fehlercharakteristika des Speicher-Pools. Somit repräsentiert ein virtuelles Gerät die Datenträger bzw. Dateien, die zum Erstellen eines Speicher-Pools verwendet werden. Ein Pool kann beliebig viele virtuelle Geräte auf oberster Konfigurationsebene aufweisen. Diese werden *vdev der obersten Ebene* genannt.

Wenn ein virtuelles Gerät der obersten Ebene zwei oder mehrere physische Geräte enthält, sorgt die Konfiguration für Datenredundanz durch Datenspiegelung oder virtuelle RAID-Z-Geräte. Diese virtuellen Geräte bestehen aus Datenträgern, Datenträgerbereichen oder Dateien. Als Spare werden spezielle virtuelle Geräte bezeichnet, die die verfügbaren Hot-Spares für einen Pool überwachen.

Das folgende Beispiel zeigt, wie ein Pool erstellt wird, der aus zwei virtuellen Geräten der obersten Ebene mit je zwei gespiegelten Festplatten besteht:

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

Das folgende Beispiel zeigt, wie ein Pool erstellt wird, der aus einem virtuellen Gerät der obersten Ebene mit vier Festplatten besteht:

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

Mit dem Befehl `zpool add` kann diesem Pool ein weiteres virtuelles Gerät der obersten Ebene hinzugefügt werden. Beispiel:

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

Festplatten, Festplattenbereiche oder Dateien, die in nicht redundanten Pools verwendet werden, fungieren als virtuelle Geräte der obersten Hierarchieebene. Speicher-Pools enthalten normalerweise mehrere virtuelle Geräte der obersten Hierarchieebene. ZFS verteilt Daten dynamisch mithilfe des sog. Stripe-Verfahrens über alle in der obersten Hierarchieebene befindlichen virtuellen Geräte eines Pools.

Virtuelle Geräte und die physischen Geräte, die in einem ZFS-Speicher-Pool enthalten sind, werden mit dem Befehl `zpool status` angezeigt. Beispiel:

```
# 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
```

Behandlung von Fehlern beim Erstellen von ZFS-Speicher-Pools

Fehler beim Erstellen von Pools können in verschiedenen Situationen auftreten. Einige Fehlergründe sind sofort offensichtlich, wenn beispielsweise ein angegebenes Gerät nicht vorhanden ist, während andere Gründe nicht so einfach nachzuvollziehen sind.

Erkennen belegter Geräte

Vor dem Formatieren eines Datenspeichergeräts versucht ZFS herauszufinden, ob dieser Datenträger bereits von ZFS oder anderen Bereichen des Betriebssystems verwendet wird. Wenn der Datenträger bereits verwendet wird, sehen Sie in etwa folgende Fehlermeldung:

```
# 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).
```

Einige Fehler können mithilfe der Option `-f` ignoriert werden, die meisten Fehler jedoch nicht. Die folgenden Zustände können mithilfe der Option `-f` nicht ignoriert werden. Sie müssen diese Zustände manuell korrigieren:

Mounted file system (eingehängtes Dateisystem)
Der Datenträger oder einige seiner Bereiche enthalten ein gegenwärtig eingehängtes Dateisystem. Sie müssen diesen Fehler mit dem Befehl `umount` beheben.

File system in /etc/vfstab (Dateisystem in /etc/vfstab)
Der Datenträger enthält ein Dateisystem, das in der Datei `/etc/vfstab` aufgeführt, jedoch gegenwärtig nicht eingehängt ist. Zur Behebung dieses Fehlers müssen Sie die entsprechende Zeile aus der Datei `/etc/vfstab` entfernen bzw. diese Zeile auskommentieren.

Dedicated dump device (reserviertes Dump-Gerät)
Dieser Datenträger ist das reservierte Dump-Gerät für dieses System. Sie müssen diesen Fehler mit dem Befehl `dumpadm` beheben.

Part of a ZFS pool (Teil eines ZFS-Pools)
Der Datenträger bzw. die Datei gehört zu einem aktiven ZFS-Speicher-Pool. Sofern nicht mehr benötigt, entfernen Sie den anderen Pool mit dem Befehl `zpool destroy`. Oder verwenden Sie den Befehl `zpool detach`, um die Festplatte aus dem anderen Pool zu trennen. Sie können nur eine Festplatte aus einem Speicher-Pool mit Datenspiegelung trennen.

Die folgenden Überprüfungen auf Datenträgerbelegungen dienen als nützliche Warnungen und können mithilfe der Option `-f` manuell übergangen werden, um den Pool zu erstellen:

Contains a file system (Enthält ein Dateisystem)
Der Datenträger enthält ein bekanntes Dateisystem, obwohl es nicht eingehängt ist und nicht in Gebrauch zu sein scheint.

Part of volume (Teil eines Volumes)
Der Datenträger gehört zu einem Solaris Volume Manager-Volume.

Live Upgrade
Der Datenträger dient als alternative Boot-Umgebung für Oracle Solaris Live Upgrade.

Part of a exported ZFS pool (Teil eines exportierten ZFS-Pools)
Der Datenträger gehört zu einem Speicher-Pool, das exportiert bzw. manuell aus einem System entfernt wurde. Im letzteren Fall wird der Pool als `potenziell aktiv` gemeldet, da der Datenträger unter Umständen über das Netzwerk von einem anderen System belegt sein kann. Überprüfen Sie den Status vor beim Überschreiben eines potenziell aktiven Datenträgers äußerst sorgfältig.

Das folgende Beispiel zeigt die Verwendung der Option `-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
```

Im Normalfall sollten Sie Fehler beheben, anstatt sie mit Option `-f` zu ignorieren.

Inkongruente Replikationsmethoden

Die Erstellung von Pools mit unterschiedlichen Replikationsmethoden wird nicht empfohlen. Der Befehl `zpool` verhindert, dass Sie versehentlich einen Pool mit inkongruenten Redundanzebenen erstellen. Wenn Sie versuchen, einen Pool mit einer solchen Konfiguration zu erstellen, wird in etwa die folgende Fehlermeldung ausgegeben:

```
# 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
```

Sie können diese Fehler mit der Option `-f` ignorieren, obwohl dies nicht empfohlen wird. Der Befehl warnt Sie darüber hinaus auch, wenn Sie versuchen, einen Pool mit Datenspiegelung bzw. RAID-Z zu erstellen, das Datenträger unterschiedlicher Kapazität verwendet. Obwohl eine solche Konfiguration zulässig ist, führen inkongruente Redundanzebenen dazu, dass die Kapazität größerer Datenträger ungenutzt bleibt. Die Option `-f` wird benötigt, um die Warnung zu ignorieren.

Ausführen eines Testlaufs für die Erstellung eines Speicher-Pools

Versuche, einen Pool zu erstellen, können unerwartet und aus verschiedenen Gründen fehlschlagen, und das Formatieren von Datenträgern ist ein potentiell schädlicher Vorgang. Aus diesen Gründen verfügt der Befehl `zpool create` über eine zusätzliche Option (`-n`), mit der das Erstellen eines Pools simuliert wird, ohne das Gerät tatsächlich zu beschreiben. Mit der Option *dry run* wird die Belegung von Datenträgern sowie die Evaluierung der Replikationsmethoden überprüft, und es werden im Verlauf dieses Vorgangs eventuell auftretende Fehler gemeldet. Wenn keine Fehler gefunden wurden, wird in etwa die folgende Meldung ausgegeben:

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

        tank
          mirror
            c1t0d0
            c1t1d0
```

Einige Fehler werden jedoch erst angezeigt, wenn der Pool tatsächlich erstellt wird. Ein Beispiel für einen solchen häufig auftretenden Fehler ist die mehrmalige Angabe eines Datenträgers in der gleichen Konfiguration. Dieser Fehler kann erst beim tatsächlichen Schreiben von Daten

zuverlässig erkannt werden. Aus diesem Grunde kann es sein, dass mit dem Befehl `zpool create -n` eine erfolgreiche Ausführung gemeldet wird, aber der Pool nicht erstellt werden kann, wenn der Befehl ohne diese Option ausgeführt wird.

Standard-Einhängepunkt für Speicher-Pools

Bei der Erstellung eines Speicherpools ist */pool-name* der Standard-Einhängepunkt für das Dateisystem der obersten Hierarchieebene. Dieses Verzeichnis darf noch nicht vorhanden oder muss leer sein. Falls das Verzeichnis noch nicht existiert, wird es autoamtisch erstellt. Wenn das Verzeichnis leer ist, wird das Root-Dateisystem in dieses vorhandene Verzeichnis eingehängt. Mit der Option `-m` des Befehls `zpool create` können Sie Pools mit anderen Standard-Einhängepunkten erstellen. Beispiel:

```
# 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
```

Mit diesem Befehl wird der neue Pool `home` sowie das Dateisystem `home` mit dem Einhängepunkt `/export/zfs` erstellt.

Weitere Informationen zu Einhängepunkten finden Sie unter „Verwalten von ZFS-Einhängepunkten“ auf Seite 204.

Löschen von ZFS-Speicher-Pools

Pools werden mit dem Befehl `zpool destroy` gelöscht. Dieser Befehl löscht einen Pool auch dann, wenn er eingehängte Datasets enthält.

```
# zpool destroy tank
```

Achtung – Seien Sie beim Löschen von Pools äußerst vorsichtig. Stellen Sie sicher, dass der richtige Pool gelöscht wird und Sie Sicherungskopien der Daten dieses Pools erstellt haben. Falls versehentlich der falsche Pool gelöscht wurde, kann er eventuell wiederhergestellt werden. Weitere Informationen dazu finden Sie unter „Wiederherstellen gelöschter ZFS-Speicher-Pools“ auf Seite 107.

Wenn Sie einen Pool mit dem Befehl `zpool destroy` endgültig löschen, kann der Pool immer noch importiert werden, wie in „Wiederherstellen gelöschter ZFS-Speicher-Pools“ auf Seite 107 beschrieben. Dies bedeutet, dass vertrauliche Daten weiterhin auf den Festplatten verfügbar sein können, die Bestandteil des Pools waren. Wenn Sie Daten auf den Festplatten des endgültig gelöschten Pools zerstören möchten, müssen Sie eine Funktion wie die Option `analyze->purge` des `format`-Serviceprogramms auf jeder Festplatte in dem endgültig gelöschten Pool verwenden.

Löschen eines Pools mit nicht verfügbaren Geräten

Wenn ein Pool gelöscht wird, müssen Daten auf den Datenträger geschrieben werden, um anzuzeigen, dass der Pool nicht mehr verfügbar ist. Diese Statusinformationen verhindern, dass beim Durchführen von Imports diese Datenspeichergeräte als potenziell verfügbar angezeigt werden. Auch wenn Datenspeichergeräte nicht verfügbar sind, kann der Pool gelöscht werden. Die erforderlichen Statusinformationen werden in diesem Fall jedoch nicht auf den nicht verfügbaren Datenträgern gespeichert.

Nach einer entsprechenden Reparatur werden diese Datenspeichergeräte bei der Erstellung neuer Pools als *potenziell aktiv* gemeldet. Sie werden als verfügbare Geräte angezeigt, wenn Sie nach Pools suchen, die importiert werden sollen. Wenn ein Pool so viele `UNAVAIL` Geräte enthält, dass der Pool selbst als `UNAVAIL` erkannt wird (d.h. das Gerät der obersten Hierarchieebene ist `UNAVAIL`), gibt der Befehl eine Warnmeldung aus und kann ohne die Option `-f` nicht fortgesetzt werden. Diese Option ist erforderlich, da der Pool nicht geöffnet werden kann und somit nicht bekannt ist, ob dort Daten gespeichert sind. Beispiel:

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

Weitere Informationen zum Funktionsstaus von Pools und Datenspeichergeräten finden Sie unter „Ermitteln des Funktionsstatus von ZFS-Speicher-Pools“ auf Seite 94.

Weitere Informationen zum Importieren von Pools finden Sie unter „Importieren von ZFS-Speicher-Pools“ auf Seite 103.

Verwalten von Datenspeichergeräten in ZFS-Speicher-Pools

Die meisten grundlegenden Informationen zu Datenspeichergeräten sind in „Komponenten eines ZFS-Speicher-Pools“ auf Seite 45 enthalten. Nach dem Erstellen eines Pools können Sie zum Verwalten der zum Pool gehörenden Datenspeichergeräte verschiedene Aufgaben ausführen.

Hinzufügen von Datenspeichergeräten zu einem Speicher-Pool

Durch Hinzufügen eines neuen Gerätes der obersten Hierarchieebene können Sie einen Pool dynamisch um Festplattenkapazität erweitern. Die zusätzliche Festplattenkapazität steht allen im Pool enthaltenen Datasets sofort zur Verfügung. Mit dem Befehl `zpool add` fügen Sie einem Pool ein neues virtuelles Gerät hinzu. Beispiel:

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

Das Format zum Angeben dieser virtuellen Geräte entspricht dem Format für den Befehl `zpool create`. Es wird geprüft, ob die betreffenden Datenspeichergeräte belegt sind, und der Befehl kann die Redundanzebene ohne die Option `-f` nicht ändern. Der Befehl unterstützt auch die Option `-n` zum Durchführen eines Testlaufs. Beispiel:

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

Mit dieser Befehlsyntax werden die gespiegelten Geräte `c3t1d0` und `c3t2d0` zur vorhandenen Konfiguration des Pools`zeepool` hinzugefügt.

Weitere Informationen zur Überprüfung von virtuellen Geräten finden Sie unter „Erkennen belegter Geräte“ auf Seite 60.

BEISPIEL 3–1 Hinzufügen von Festplatten in eine ZFS-Konfiguration mit Datenspiegelung

Im folgenden Beispiel wird eine weitere Spiegelung der vorhandenen gespiegelten ZFS-Konfiguration hinzugefügt.

```
# 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
```

BEISPIEL 3–1 Hinzufügen von Festplatten in eine ZFS-Konfiguration mit Datenspiegelung *(Fortsetzung)*

```
            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
```

BEISPIEL 3–2 Hinzufügen von Datenträgern zu einer RAID-Z-Konfiguration

Datenträger können in ähnlicher Weise auch zu einer RAID-Z-Konfiguration hinzugefügt werden. Das folgende Beispiel zeigt, wie ein Speicher-Pool mit einem aus drei Festplatten bestehenden RAID-Z-Gerät in ein Speicher-Pool mit zwei aus je drei Datenträgern bestehenden RAID-Z-Geräten umgewandelt werden kann.

```
# 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:
```

BEISPIEL 3–2 Hinzufügen von Datenträgern zu einer RAID-Z-Konfiguration *(Fortsetzung)*

```
        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
```

BEISPIEL 3–3 Hinzufügen und Entfernen eines gespiegelten Protokolliergeräts

Im folgenden Beispiel wird dargestellt, wie Sie ein gespiegeltes Loggerät zu einem gespiegelten Speicherpool hinzufügen.

```
# 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
# 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
```

Sie können zur Datenspiegelung an ein vorhandenes Protokolliergerät ein weiteres Protokolliergerät anschließen. Dies entspricht dem Verbinden eines Speichergeräts in einem Speicher-Pool ohne Datenspiegelung.

BEISPIEL 3–3 Hinzufügen und Entfernen eines gespiegelten Protokolliergeräts *(Fortsetzung)*

Mit dem Befehl `zpool remove` können Sie Protokolliergeräte entfernen. Das gespiegelte Protokolliergerät, um das es im vorherigen Beispiel geht, kann durch Angabe des Arguments `mirror-1` entfernt werden. Beispiel:

```
# 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
```

Wenn Ihre Pool-Konfiguration nur ein Protokolliergerät enthält, entfernen Sie das Gerät, indem Sie dessen Namen angeben. Beispiel:

```
# 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
```

BEISPIEL 3–4 Hinzufügen und Entfernen von Cache-Geräten

Sie können Cachegeräte zu Ihrem ZFS-Speicherpool hinzufügen oder Cachegeräte aus Ihrem ZFS-Speicherpool entfernen, wenn diese nicht mehr benötigt werden.

Verwenden Sie den Befehl `zpool add` zum Hinzufügen von Cache-Geräten. Beispiele:

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

BEISPIEL 3–4 Hinzufügen und Entfernen von Cache-Geräten *(Fortsetzung)*

```
        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
```

Cachegeräte können nicht gespiegelt werden oder Teil einer RAID-Z-Konfiguration sein.

Verwenden Sie den Befehl `zpool remove` zum Entfernen von Cache-Geräten. Beispiel:

```
# zpool remove tank 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

errors: No known data errors
```

Der Befehl `zpool remove` unterstützt derzeit nur das Entfernen von Hot-Spares, Protokolliergeräten und Cache-Geräten. Geräte, die zur primären gespiegelten Pool-Konfiguration gehören, können mit dem Befehl `zpool detach` entfernt werden. Geräte ohne Redundanz und RAID-Z-Geräte können nicht aus einem Pool entfernt werden.

Weitere Informationen zur Verwendung von Cache-Geräten in einem ZFS-Speicher-Pool finden Sie unter „Erstellen eines ZFS-Speicher-Pools mit Cache-Geräten“ auf Seite 57.

Verbinden und Trennen von Geräten in einem Speicher-Pool

Zusätzlich zum Befehl `zpool add` können Sie mit dem Befehl `zpool attach` zu einem Gerät mit oder ohne Datenspiegelung ein neues Datenspeichergerät hinzufügen.

Wenn Sie eine Festplatte einbinden möchten, um einen gespiegelten Root-Pool zu erstellen, gehen Sie wie unter „Erstellen eines gespiegelten ZFS-Root-Pools (nach der Installation)“ auf Seite 122 beschrieben vor.

Informationen über das Ersetzen einer Festplatte in einem ZFS-Root-Pool finden Sie in „Wiederherstellen von ZFS-Root-Pools oder Root-Pool-Snapshots“ auf Seite 169.

BEISPIEL 3–5 Umwandlung eines Speicher-Pools mit zweifacher Datenspiegelung in einen Speicher-Pool mit dreifacher Datenspiegelung

In diesem Beispiel ist `zeepool` eine vorhandene zweifache Datenspiegelungskonfiguration, die durch Verbinden des Geräts `c2t1d0` mit dem vorhandenen Gerät `c1t1d0` in eine dreifache Datenspiegelungskonfiguration umgewandelt wird.

```
# 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
            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
```

Wenn das betreffende Gerät zu einer dreifachen Datenspiegelungskonfiguration gehört, wird durch Verbinden des neuen Geräts eine vierfache Datenspiegelungskonfiguration erstellt usw. In jedem Fall wird sofort das Resilvering durchgeführt (d. h., die gespiegelten Daten werden auf das neue Gerät in der Datenspiegelungskonfiguration übertragen).

BEISPIEL 3–6 Umwandeln eines ZFS-Speicher-Pools ohne Redundanz in einen ZFS-Speicher-Pool mit Datenspiegelung

Außerdem können Sie mithilfe des Befehls `zpool attach` einen Speicher-Pool ohne Redundanz in einen Pool mit Redundanz umwandeln. Beispiel:

```
# zpool create tank c0t1d0
# zpool status tank
  pool: tank
 state: ONLINE
```

BEISPIEL 3–6 Umwandeln eines ZFS-Speicher-Pools ohne Redundanz in einen ZFS-Speicher-Pool mit Datenspiegelung *(Fortsetzung)*

```
 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
        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
```

Mit dem Befehl `zpool detach` kann ein Datenspeichergerät aus einem Pool mit Datenspiegelungskonfiguration abgetrennt werden. Beispiel:

```
# zpool detach zeepool c2t1d0
```

Dieser Vorgang ist allerdings nicht ausführbar, wenn keine anderen Replikationen der betreffenden Daten vorhanden sind. Beispiel:

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

Erstellen eines neuen Pools durch Teilen eines ZFS-Speicher-Pools mit Datenspiegelung

Ein gespiegelter ZFS-Speicherpool kann schnell mit dem Befehl `zpool split` als Sicherungspool geklont werden. Sie können diese Funktion verwenden, um einen gespiegelten Root-Pool zu teilen, der geteilte Pool kann jedoch erst gebootet werden, nachdem Sie einige zusätzliche Schritte ausgeführt haben.

Mithilfe des Befehls `zpool split` können Sie eine oder mehrere Festplatten von einem ZFS-Speicher-Pool mit Datenspiegelung trennen, um einen neuen Pool mit der/den getrennten Festplatte(n) zu erstellen. Der Inhalt des neuen Pools ist mit dem des ursprünglichen ZFS-Speicher-Pools mit Datenspiegelung identisch.

Durch den Befehl `zpool split` wird die letzte Festplatte eines Pools mit Datenspiegelung standardmäßig getrennt und für den neuen Pool verwendet. Nach der Teilung importieren Sie den neuen Pool. Beispiel:

```
# 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
```

Sie können feststellen, welche Festplatte für den neuen Pool verwendet werden soll, indem Sie sie mithilfe des Befehls `zpool split` angeben. Beispiel:

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

Vor der tatsächlichen Teilung werden die im Speicher enthaltenen Daten auf die gespiegelten Festplatten ausgespeichert. Nach dem Ausspeichern wird die Festplatte vom Pool getrennt und erhält eine neue Pool-GUID. Die neue Pool-GUID wird generiert, damit der Pool auf dasselbe System importiert werden kann, auf dem er geteilt wurde.

Wenn der Pool, der geteilt werden soll, keine standardmäßigen Dateisystem-Einhängepunkte hat und der neue Pool auf demselben System erstellt wird, müssen Sie die Option `zpool split`

`-R` verwenden, um ein alternatives Root-Verzeichnis für den neuen Pool zu bestimmen, damit es nicht zu Konflikten zwischen vorhandenen Einhängepunkten kommt. Beispiel:

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

Wenn Sie nicht die Option `zpool split -R` verwenden und feststellen, dass Einhängepunkte in Konflikt geraten, sobald Sie versuchen, den neuen Pool zu importieren, importieren Sie den neuen Pool mithilfe der Option `-R`. Wenn der neue Pool auf einem anderen System erstellt wird, ist die Angabe eines alternativen Root-Verzeichnisses nicht nötig, es sei denn, es treten Einhängepunkt-Konflikte auf.

Berücksichtigen Sie Folgendes, bevor Sie die Funktion `zpool split` verwenden:

- Diese Funktion steht für RAID-Z-Konfigurationen oder nicht redundante Pools mit mehreren Festplatten nicht zur Verfügung.
- Datenverarbeitungs- und Anwendungsprozesse sollten unterbrochen werden, bevor die Funktion `zpool split` verwendet wird.
- Ein Pool kann nicht geteilt werden, wenn das Resilvering im Gange ist.
- Ein Pool mit Datenspiegelung kann optimal geteilt werden, wenn dieser zwei bis drei Festplatten enthält und die letzte Festplatte des Pools für den neuen Pool verwendet wird. Anschließend können Sie den Befehl `zpool attach` verwenden, um den ursprünglichen Speicher-Pool mit Datenspiegelung wiederherzustellen oder den neuen Pool in einem Speicher-Pool mit Datenspiegelung umzuwandeln. Aktuell gibt es keine Methode für das Erstellen eines *neuen* gespiegelten Pools aus einem *vorhandenen* gespiegelten Pool in einem `zpool split`-Vorgang, weil der neue (geteilte) Pool nicht redundant ist
- Wenn der vorhandene Pool ein Pool mit dreifacher Datenspiegelung ist, enthält der neue Pool nach der Teilung eine Festplatte. Wenn der vorhandene Pool ein Pool mit zweifacher Datenspiegelung ist und zwei Festplatten enthält, entstehen durch die Teilung zwei nicht redundante Pools mit zwei Festplatten. Sie müssen zwei weitere Festplatten einbinden, um die nicht redundanten Pools in Pools mit Datenspiegelung umzuwandeln.
- Ein gute Methode zur Beibehaltung der Daten während einer Teilung ist es, einen Speicher-Pool mit Datenspiegelung, der drei Festplatten enthält, zu teilen, sodass der ursprüngliche Pool nach der Teilung zwei Festplatten mit Datenspiegelung enthält.
- Vergewissern Sie sich, dass die Hardware ordnungsgemäß konfiguriert ist, bevor Sie einen gespiegelten Pool teilen. Informationen zur Prüfung der Cache-Flush-Einstellungen der Hardware finden Sie in „Allgemeine Vorgehensweise bei dem System“ auf Seite 331.

BEISPIEL 3–7 Teilung eines ZFS-Speicher-Pools mit Datenspiegelung

Im folgenden Beispiel wird ein gespiegelter Speicherpool namens `mothership` mit drei Festplatten geteilt. Daraus ergeben sich die beiden folgenden Pools: Der gespiegelte Pool `mothership` mit zwei Festplatten und der neue Pool, `luna` , mit einer Festplatte. Der Inhalt beider Pools ist identisch.

BEISPIEL 3–7 Teilung eines ZFS-Speicher-Pools mit Datenspiegelung *(Fortsetzung)*

Der Pool `luna` könnte zu Backup-Zwecken in ein anderes System importiert werden. Nach Abschluss des Backups könnte der Pool `luna` endgültig gelöscht werden und die Festplatte könnte wieder `mothership` zugeordnet werden. Danach könnte der Prozess wiederholt werden.

```
# zpool status mothership
  pool: mothership
 state: ONLINE
  scan: none requested
config:

        NAME                        STATE     READ WRITE CKSUM
        mothership                  ONLINE       0     0     0
          mirror-0                  ONLINE       0     0     0
            c0t5000C500335F95E3d0   ONLINE       0     0     0
            c0t5000C500335BD117d0   ONLINE       0     0     0
            c0t5000C500335F907Fd0   ONLINE       0     0     0

errors: No known data errors
# zpool split mothership luna
# zpool import luna
# zpool status mothership luna
  pool: luna
 state: ONLINE
  scan: none requested
config:

        NAME                      STATE     READ WRITE CKSUM
        luna                      ONLINE       0     0     0
          c0t5000C500335F907Fd0   ONLINE       0     0     0

errors: No known data errors

  pool: mothership
 state: ONLINE
  scan: none requested
config:

        NAME                        STATE     READ WRITE CKSUM
        mothership                  ONLINE       0     0     0
          mirror-0                  ONLINE       0     0     0
            c0t5000C500335F95E3d0   ONLINE       0     0     0
            c0t5000C500335BD117d0   ONLINE       0     0     0

errors: No known data errors
```

In- und Außerbetriebnehmen von Geräten in einem Speicher-Pool

Einzelne Datenspeichergeräte können in ZFS in und außer Betrieb genommen werden. Wenn Hardwarekomponenten unzuverlässig sind und nicht richtig funktionieren, schreibt ZFS trotzdem weiter Daten auf das betreffende Gerät bzw. liest sie von diesem und geht davon aus,

dass dieser Zustand nicht von Dauer ist. Wenn dieser Zustand andauert, können Sie ZFS anweisen, das Gerät zu ignorieren, wodurch es außer Betrieb genommen wird. ZFS sendet dann keine Anforderungen an das außer Betrieb genommene Gerät.

Hinweis – Zum Austauschen von Datenspeichergeräten müssen diese vorher nicht außer Betrieb genommen werden.

Außerbetriebnehmen eines Geräts

Datenspeichergeräte werden mit dem Befehl `zpool offline` außer Betrieb genommen. Wenn es sich bei dem Datenspeichergerät um eine Festplatte handelt, kann es mit dem vollständigen Pfad oder einer Kurzbezeichnung angegeben werden. Beispiel:

```
# zpool offline tank c0t5000C500335F95E3d0
```

Beim Außerbetriebnehmen von Datenspeichergeräten sollten Sie Folgendes berücksichtigen:

- Pools können nicht außer Betrieb genommen werden, wenn sie dadurch `UNAVAIL` werden. So können Sie beispielsweise zwei Geräte nicht aus einer `raids1`-Konfiguration oder ein virtuelles Gerät der obersten Hierarchieebene nicht außer Betrieb nehmen.

  ```
  # zpool offline tank c0t5000C500335F95E3d0
  cannot offline c0t5000C500335F95E3d0: no valid replicas
  ```

- Der `OFFLINE`-Status ist dauerhaft, d. h. das Gerät bleibt auch bei einem Systemneustart außer Betrieb.

  Mit dem Befehl `zpool offline -t` können Sie ein Datenspeichergerät zeiweilig außer Betrieb nehmen. Beispiel:

  ```
  # zpool offline -t tank c1t0d0
  ```

  Bei einem Systemneustart wird dieses Gerät dann automatisch in den `ONLINE`-Status gebracht.

- Wenn ein Gerät außer Betrieb genommen wurde, wird es nicht automatisch vom Speicher-Pool getrennt. Wenn Sie versuchen, das betreffende Gerät für einen anderen Pool zu verwenden (selbst wenn dieser Pool gelöscht wurde), wird eine Meldung wie die folgende angezeigt:

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

  Wenn Sie das außer Betrieb genommene Gerät für einen anderen Pool verwenden möchten, nachdem der ursprüngliche Pool gelöscht wurde, müssen Sie zunächst das Gerät wieder in Betrieb nehmen und dann den ursprünglichen Pool löschen.

  Eine andere Methode, ein Gerät eines anderen Speicher-Pools zu nutzen, während der ursprüngliche Pool beibehalten wird, besteht darin, das vorhandene Gerät im ursprünglichen Pool durch ein anderes, vergleichbares Gerät zu ersetzen. Weitere Informationen zum Austauschen von Geräten finden Sie unter „Austauschen von Geräten in einem Speicher-Pool“ auf Seite 77.

Wenn Sie den Pool-Status abfragen, weisen außer Betrieb genommene Geräte den Status `OFFLINE` auf. Weitere Informationen zum Abfragen des Pool-Status finden Sie unter „Abfragen des Status von ZFS-Speicher-Pools“ auf Seite 88.

Weitere Informationen zum Funktionsstatus von Datenspeichergeräten finden Sie unter „Ermitteln des Funktionsstatus von ZFS-Speicher-Pools“ auf Seite 94.

Inbetriebnehmen eines Gerätes

Nach dem Außerbetriebnehmen eines Geräts kann es mit dem Befehl `zpool online` wieder in Betrieb genommen werden. Beispiel:

```
# zpool online tank c0t5000C500335F95E3d0
```

Nach dem Inbetriebnehmen des Gerätes werden alle Daten, die im Pool gespeichert wurden, mit dem neu verfügbaren Gerät synchronisiert. Beachten Sie, dass Sie eine Festplatte nicht ersetzen können, indem Sie ein Gerät in Betrieb nehmen. Wenn Sie ein Gerät außer Betrieb nehmen, es austauschen und versuchen, es wieder in Betrieb zu nehmen, bleibt es im Status `UNAVAIL`.

Wenn Sie versuchen, ein `UNAVAIL`-Gerät in Betrieb zu nehmen, wird eine Meldung wie die Folgende angezeigt:

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

Die Fehlermeldung kann auch an der Konsole angezeigt oder in die Datei unter `/var/adm/messages` geschrieben werden. Beispiel:

```
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Wed Jun 20 11:35:26 MDT 2012
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.
```

Weitere Informationen zum Austauschen fehlerhafter Geräte finden Sie unter „Abhilfe bei fehlendem oder entferntem Gerät“ auf Seite 303.

Mit dem Befehl `zpool online -e` können Sie die Pool-Größe erweitern, wenn eine größere Festplatte in den Pool eingebunden oder eine kleinere Festplatte durch eine größere ersetzt wurde. Eine Festplatte, die zu einem Pool hinzugefügt wird, wird standardmäßig nicht auf ihre volle Größe erweitert, wenn die Pool-Eigenschaft `autoexpand` nicht aktiviert ist. Mithilfe des Befehls `zpool online -e` können Sie die Festplatte automatisch erweitern, selbst dann, wenn die Ersatzfestplatte bereits aktiviert wurde oder wenn die Festplatte gerade deaktiviert ist. Beispiel:

```
# zpool online -e tank c0t5000C500335F95E3d0
```

Löschen von Gerätefehlern im Speicher-Pool

Wenn ein Datenspeichergerät aufgrund von Fehlern, die in der Ausgabe von `zpool status` aufgeführt werden, außer Betrieb genommen wird, können Sie diese Fehler mithilfe des Befehls `zpool clear` löschen.

Wenn keine Argumente angegeben werden, löscht der Befehl alle Gerätefehler eines Pools. Beispiel:

```
# zpool clear tank
```

Wenn mindestens ein Gerät angegeben wird, löscht der Befehl nur die zu den betreffenden Geräten gehörenden Fehler. Beispiel:

```
# zpool clear tank c0t5000C500335F95E3d0
```

Weitere Informationen zum Löschen von `zpool`-Fehlern finden Sie unter „Zurücksetzen vorübergehender Gerätefehler“ auf Seite 309.

Austauschen von Geräten in einem Speicher-Pool

Mit dem Befehl `zpool replace` können Sie Datenspeichergeräte in einem Speicher-Pool austauschen.

Wenn Sie in einem Pool mit Redundanz ein Datenspeichergerät an der gleichen Stelle durch ein anderes Datenspeichergerät ersetzen, brauchen Sie nur das ersetzte Datenspeichergerät anzugeben. Bei mancher Hardware erkennt ZFS, dass das Gerät eine andere Festplatte ist, die sich an derselben Stelle befindet. Wenn Sie beispielsweise eine ausgefallene Festplatte (`c1t1d0`) durch Auswechseln an der gleichen Stelle ersetzen wollen, verwenden Sie folgende Syntax:

```
# zpool replace tank c1t1d0
```

Wenn Sie ein Gerät in einem Speicher-Pool durch eine Festplatte an einer anderen physischen Stelle ersetzen wollen, müssen Sie beide Geräte angeben. Beispiel:

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

Informationen über das Ersetzen einer Festplatte im ZFS-Root-Pool finden Sie in „Wiederherstellen von ZFS-Root-Pools oder Root-Pool-Snapshots“ auf Seite 169.

Es folgen die grundlegenden Schritte zum Austauschen von Datenträgern:

1. Nehmen Sie den Datenträger wenn nötig mit dem Befehl `zpool offline` außer Betrieb.
2. Bauen Sie die zu erstetzende Festplatte aus.

3. Setzen Sie die Ersatzfestplatte ein.
4. Prüfen Sie anhand der `format`-Ausgabe, ob die Ersatzfestplatte sichtbar ist.

   Prüfen Sie außerdem, ob die Geräte-ID geändert wurde. Wenn die Ersatzfestplatte über eine WWN verfügt, wurde die Geräte-ID für die fehlerhafte Festplatte geändert.
5. Informieren Sie ZFS, dass die Festplatte ersetzt wurde. Beispiel:

   ```
   # zpool replace tank c1t1d0
   ```

   Wenn die Ersatzfestplatte eine andere Geräte-ID als die oben angegebene ID hat, beziehen Sie die neue Geräte-ID ein.

   ```
   # zpool replace tank c0t5000C500335FC3E7d0 c0t5000C500335BA8C3d0
   ```
6. Nehmen Sie den Datenträger falls erforderlich mit dem Befehl `zpool online` in Betrieb.
7. Benachrichtigen Sie FMA, dass das Gerät ersetzt wurde.

   Identifizieren Sie anhand der `fmadm faulty`-Ausgabe die Zeichenfolge `zfs://pool=name/vdev=guid` im Abschnitt `Affects:`, und geben Sie diese Zeichenfolge als Argument für den Befehl `fmadm repaired` an.

   ```
   # fmadm faulty
   # fmadm repaired zfs://pool=name/vdev=guid
   ```

Bei einigen Systemen mit SATA-Festplatten müssen Sie die Festplatte dekonfigurieren, bevor Sie sie offline setzen. Wenn Sie bei diesem System eine Festplatte an ein und demselben Steckplatz austauschen, genügt es, den Befehl `zpool replace` wie im ersten Beispiel dieses Abschnitt beschrieben auszuführen.

Ein Beispiel für das Ersetzen einer SATA-Festplatte finden Sie in Beispiel 10–1.

Beachten Sie beim Auswechseln von Datenspeichergeräten in einem ZFS-Speicher-Pool Folgendes:

- Wenn Sie die Eigenschaft `autoreplace` des Pools auf `on` setzen, wird jedes neue Datenspeichergerät, das sich an der physischen Stelle eines zuvor zum Pool gehörenden Datenspeichergeräts befindet, automatisch formatiert und ersetzt. Der Befehl `zpool replace` muss nicht verwendet werden, wenn diese Eigenschaft aktiviert ist. Dieses Leistungsmerkmal ist möglicherweise nicht auf jeder Art von Hardware verfügbar.
- Wenn ein Speichergerät oder Hot-Spare bei laufendem Betrieb physisch entfernt wurde, steht jetzt der Speicherpoolstatus `REMOVED` zur Verfügung. Falls verfügbar, wird ein Hot-Spare für das entfernte Speichergerät in den Pool eingebunden.
- Wenn ein Speichergerät entfernt und danach wieder eingesetzt wird, wird es in Betrieb genommen. Wenn für das entfernte Speichergerät ein Hot-Spare eingebunden wurde, wird dieses nach Abschluss der Inbetriebnahme wieder entfernt.

- Das automatische Erkennen entfernter und hinzugefügter Speichergeräte ist hardwareabhängig und wird nicht von allen Plattformen unterstützt. So werden USB-Speichergeräte beispielsweise beim Einfügen automatisch konfiguriert. SATA-Laufwerke müssen unter Umständen jedoch mit dem Befehl `cfgadm -c configure` konfiguriert werden.
- Hot-Spares werden regelmäßig überprüft, um sicherzustellen, dass sie in Betrieb und verfügbar sind.
- Die Kapazität des Austauschgeräts muss der Kapazität der kleinsten Festplatte in einer Datenspiegelungs- bzw. RAID-Z-Konfiguration entsprechen oder größer sein.
- Wenn ein Austauschgerät – dessen Kapazität größer ist als die des Geräts, das ausgetauscht wird – zu einem Pool hinzugefügt wird, wird es nicht automatisch auf seine volle Kapazität erweitert. Der Pooleigenschaftswert `autoexpand` bestimmt, ob der Pool erweitert wird, wenn ihm eine größere Festplatte hinzugefügt wird. Standardmäßig ist die Eigenschaft `autoexpand` aktiviert. Sie können diese Eigenschaft aktivieren, um die Pool-Größe zu erweitern, bevor oder nachdem die größere Festplatte zum Pool hinzugefügt wird.

  Im folgenden Beispiel werden zwei 16-GB-Festplatten in einem Pool mit Datenspiegelung durch zwei 72-GB-Festplatten ersetzt. Stellen Sie sicher, dass das erste Gerät vollständig wiederhergestellt ist, bevor Sie versuchen, das zweite Gerät zu ersetzen. Nach dem Ersetzen der Festplatten wird die Eigenschaft `autoexpand` aktiviert, um die Festplatten auf die volle Größe zu erweitern.

```
# 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  -
```

- Das Austauschen mehrerer Festplatten in einem großen Pool ist zeitaufwändig, da die Daten mithilfe von Resilvering auf die neuen Festplatten aufgespielt werden müssen. Außerdem sollten Sie zwischen dem Austausch von Festplatten den Befehl `zpool scrub` ausführen, um sicherzustellen, dass die Austauschgeräte ordnungsgemäß funktionieren und Daten fehlerfrei geschrieben werden.
- Wenn eine ausgefallene Festplatte automatisch durch eine Hot-Spare-Festplatte ersetzt wurde, müssen Sie die Hot-Spare-Festplatte möglicherweise nach dem Ersetzen der ausgefallenen Festplatte abtrennen. Mit dem Befehl `zpool detach` kann eine Ersatzfestplatte in einem gespiegelten oder RAID-Z-Pool abgetrennt werden. Weitere Informationen zum Abtrennen von Hot-Spares finden Sie unter „Aktivieren und Deaktivieren von Hot-Spares im Speicher-Pool“ auf Seite 82.

Weitere Informationen zum Austauschen von Geräten finden Sie unter „Abhilfe bei fehlendem oder entferntem Gerät“ auf Seite 303 sowie „Ersetzen oder Reparieren eines beschädigten Geräts“ auf Seite 307.

Zuweisen von Hot-Spares im Speicher-Pool

Mit der Hot-Spare-Funktion können Sie Datenträger ermitteln, die zum Ersetzen eines ausgefallenen bzw. fehlerhaften Geräts in einem Speicherpool verwendet werden können. Das Vorsehen eines Geräts als *Hot-Spare* bedeutet, dass das Gerät im Pool nicht aktiv ist, sondern nur dazu dient, ein ausgefallenes Gerät im Pool automatisch zu ersetzen.

Datenspeichergeräte können mit den folgenden Methoden als Hot-Spares vorgesehen werden:

- bei der Erstellung eines Pools mit dem Befehl `zpool create`
- nach der Erstellung eines Pools mit dem Befehl `zpool add`

Das folgende Beispiel zeigt, wie Geräte bei der Erstellung des Pools als Hot-Spares zugewiesen werden:

```
# zpool create zeepool mirror c0t5000C500335F95E3d0 c0t5000C500335F907Fd0
mirror c0t5000C500335BD117d0 c0t5000C500335DC60Fd0 spare c0t5000C500335E106Bd0 c0t5000C500335FC3E7d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
  scan: none requested
config:

        NAME                       STATE     READ WRITE CKSUM
        zeepool                    ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0
        spares
```

```
          c0t5000C500335E106Bd0    AVAIL
          c0t5000C500335FC3E7d0    AVAIL

errors: No known data errors
```

Das folgende Beispiel zeigt, wie Hot-Spares zugewiesen werden, indem sie zu einem Pool hinzugefügt werden, nachdem dieser erstellt wurde:

```
# zpool add zeepool spare c0t5000C500335E106Bd0 c0t5000C500335FC3E7d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
  scan: none requested
config:

        NAME                       STATE     READ WRITE CKSUM
        zeepool                    ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0
        spares
          c0t5000C500335E106Bd0    AVAIL
          c0t5000C500335FC3E7d0    AVAIL

errors: No known data errors
```

Hot-Spares können mit dem Befehl `zpool remove` aus einem Speicher-Pool entfernt werden. Beispiel:

```
# zpool remove zeepool c0t5000C500335FC3E7d0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
  scan: none requested
config:

        NAME                       STATE     READ WRITE CKSUM
        zeepool                    ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0
        spares
          c0t5000C500335E106Bd0    AVAIL

errors: No known data errors
```

Ein Hot-Spare kann nur entfernt werden, wenn es gerade nicht vom Speicher-Pool verwendet wird.

Beachten Sie beim Arbeiten mit ZFS-Hot-Spares Folgendes:

- Zurzeit können mit dem Befehl `zpool remove` nur Hot-Spares, Cache-Geräte und Protokolliergeräte entfernt werden.
- Wenn eine Festplatte als Hot-Spare hinzugefügt werden soll, muss die Kapazität des Hot-Spares der Kapazität der größten Festplatte im Pool entsprechen oder größer sein. Sie können Festplatten mit geringerer Kapazität zwar hinzufügen, sollten jedoch beachten, dass bei Aktivierung der Festplatte mit der geringeren Kapazität (automatisch oder mithilfe des Befehls `zpool replace`) der Vorgang mit einer Fehlermeldung wie der Folgenden abbricht:

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

Aktivieren und Deaktivieren von Hot-Spares im Speicher-Pool

Hot-Spares können mit einer der folgenden Methoden aktiviert werden:

- Manueller Austausch – Mithilfe des Befehls `zpool replace` können Sie ein ausgefallenes Gerät im Speicher-Pool durch ein Hot-Spare ersetzen.
- Automatischer Austausch – Wenn ein Fehler erkannt wird, überprüft ein FMA-Agent den Pool, um festzustellen, ob dieser verfügbare Hot-Spares enthält. Wenn dies der Fall ist, ersetzt er das fehlerhafte Gerät durch ein verfügbares Ersatzgerät.

  Falls ein gerade verwendetes Hot-Spare ausfällt, trennt der FMA-Agent das Ersatzgerät ab und macht den Austausch somit rückgängig. Dann versucht der Agent, das Datenspeichergerät mit einem anderen Hot-Spare zu ersetzen, falls dies möglich ist. Diese Funktion ist gegenwärtig Beschränkungen unterworfen, da das ZFS-Diagnoseprogramm nur Fehler generiert, wenn ein Datenspeichergerät aus dem System verschwindet.

  Wenn Sie ein ausgefallenes Datenspeichergerät physisch durch ein aktives Ersatzgerät ersetzen, können Sie das ursprüngliche Gerät wieder aktivieren, indem Sie das Ersatzgerät mithilfe des Befehls `zpool detach` abtrennen. Wenn Sie die Eigenschaft `autoreplace` des Pools auf `on` setzen, wird das Ersatzgerät automatisch getrennt und wieder in den Ersatzgeräte-Pool zurückgeführt, sobald das neue Datenspeichergerät eingesetzt und die Inbetriebnahme abgeschlossen ist.

Ein `UNAVAIL`-Datenspeichergerät wird automatisch ausgetauscht, wenn ein Hot-Spare verfügbar ist. Beispiel:

```
# zpool status -x
  pool: zeepool
 state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
        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
  scan: resilvered 3.15G in 0h0m with 0 errors on Mon Nov 12 15:53:42 2012
config:

        NAME                        STATE     READ WRITE CKSUM
        zeepool                     DEGRADED     0     0     0
          mirror-0                  ONLINE       0     0     0
```

```
            c0t5000C500335F95E3d0    ONLINE       0     0     0
            c0t5000C500335F907Fd0    ONLINE       0     0     0
          mirror-1                   DEGRADED     0     0     0
            c0t5000C500335BD117d0    ONLINE       0     0     0
            spare-1                  DEGRADED   449     0     0
              c0t5000C500335DC60Fd0  UNAVAIL      0     0     0
              c0t5000C500335E106Bd0  ONLINE       0     0     0
        spares
          c0t5000C500335E106Bd0      INUSE

errors: No known data errors
```

Derzeit können Sie ein Hot-Spare wie folgt deaktivieren:

- Durch Entfernen des Hot-Spares aus dem Speicher-Pool.
- Durch Trennen eines Hot-Spares, nachdem eine ausgefallene Festplatte physisch ersetzt wurde. Siehe Beispiel 3–8.
- Durch temporäres oder dauerhaftes "Einlagern" (Swap-in) eines Hot-Spares. Siehe Beispiel 3–9.

BEISPIEL 3–8 Trennen eines Hot-Spares, nachdem eine ausgefallene Festplatte physisch ersetzt wurde

In diesem Beispiel wird die ausgefallene Festplatte (`c0t5000C500335DC60Fd0`) physisch ersetzt, und ZFS wird benachrichtigt. Dazu wird der Befehl `zpool replace` verwendet.

```
# zpool replace zeepool c0t5000C500335DC60Fd0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
  scan: resilvered 3.15G in 0h0m with 0 errors on Thu Jun 21 16:53:43 2012
config:

        NAME                       STATE     READ WRITE CKSUM
        zeepool                    ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0
        spares
          c0t5000C500335E106Bd0    AVAIL
```

Falls erforderlich können Sie den Befehl `zpool detach` verwenden, um das Hot-Spare in den Ersatzgerätepool zurückzuführen. Beispiel:

```
# zpool detach zeepool c0t5000C500335E106Bd0
```

BEISPIEL 3–9 Trennen einer ausgefallenen Festplatte und Verwenden des Hot-Spares

Wenn Sie eine ausgefallene Festplatte durch temporäres oder dauerhaftes Einlagern des Hot-Spares, das die ausgefallene Festplatte gerade ersetzt, ersetzen möchten, trennen Sie die

BEISPIEL 3–9 Trennen einer ausgefallenen Festplatte und Verwenden des Hot-Spares *(Fortsetzung)*

ursprüngliche (ausgefallene) Festplatte. Wenn die ausgefallene Festplatte schließlich ersetzt ist, können Sie sie wieder in den Speicher-Pool zurückführen, wo sie als Ersatzfestplatte verfügbar ist. Beispiel:

```
# zpool status zeepool
  pool: zeepool
 state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
        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
  scan: resilver in progress since Mon Nov 12 16:04:12 2012
     4.80G scanned out of 12.0G at 55.8M/s, 0h2m to go
     4.80G scanned out of 12.0G at 55.8M/s, 0h2m to go
     4.77G resilvered, 39.97% done
config:

        NAME                       STATE     READ WRITE CKSUM
        zeepool                    DEGRADED     0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 DEGRADED     0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  UNAVAIL      0     0     0
        spares
          c0t5000C500335E106Bd0    AVAIL

errors: No known data errors
# zpool detach zeepool c0t5000C500335DC60Fd0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
  scan: resilvered 11.3G in 0h3m with 0 errors on Mon Nov 12 16:07:12 2012
config:

        NAME                       STATE     READ WRITE CKSUM
        zeepool                    ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                 ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335E106Bd0  ONLINE       0     0     0

errors: No known data errors
(Original failed disk c0t5000C500335DC60Fd0 is physically replaced)
# zpool add zeepool spare c0t5000C500335DC60Fd0
# zpool status zeepool
  pool: zeepool
 state: ONLINE
  scan: resilvered 11.2G in 0h3m with 0 errors on Mon Nov 12 16:07:12 2012

config:
```

BEISPIEL 3–9 Trennen einer ausgefallenen Festplatte und Verwenden des Hot-Spares *(Fortsetzung)*

```
        NAME                      STATE     READ WRITE CKSUM
        zeepool                   ONLINE       0     0     0
          mirror-0                ONLINE       0     0     0
            c0t5000C500335F95E3d0 ONLINE       0     0     0
            c0t5000C500335F907Fd0 ONLINE       0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000C500335BD117d0 ONLINE       0     0     0
            c0t5000C500335E106Bd0 ONLINE       0     0     0
        spares
          c0t5000C500335DC60Fd0   AVAIL

errors: No known data errors
```

Nachdem eine Festplatte ersetzt und die Spare-Festplatte getrennt wurde, benachrichtigen Sie FMA, dass die Festplatte repariert wurde.

```
# fmadm faulty
# fmadm repaired zfs://pool=name/vdev=guid
```

Eigenschaften von ZFS-Speicher-Pools

Mit dem Befehl `zpool get` können Sie Informationen zu den Pool-Eigenschaften abrufen. Beispiel:

```
# zpool get all tank
tank  size           68G                  -
tank  capacity       0%                   -
tank  altroot        -                    default
tank  health         ONLINE               -
tank  guid           15560293364730146756 -
tank  version        32                   default
tank  bootfs         -                    default
tank  delegation     on                   default
tank  autoreplace    off                  default
tank  cachefile      -                    default
tank  failmode       wait                 default
tank  listsnapshots  on                   default
tank  autoexpand     off                  default
tank  free           68.0G                -
tank  allocated      124K                 -
tank  readonly       off                  -
```

Mit dem Befehl `zpool set` lassen sich die Pool-Eigenschaften festlegen. Beispiel:

```
# zpool set autoreplace=on zeepool
# zpool get autoreplace zeepool
NAME     PROPERTY     VALUE    SOURCE
zeepool  autoreplace  on       local
```

Wenn Sie versuchen, eine Pooleigenschaft in einem Pool festzulegen, der 100 % voll ist, wird eine Meldung wie die Folgende angezeigt.

```
# zpool set autoreplace=on tank
cannot set property for 'tank': out of space
```

Weitere Informationen, wie Probleme mit der Speicherkapazität des Pools vermieden werden, finden Sie in Kapitel 11, „Empfohlene Oracle Solaris ZFS-Vorgehensweisen“.

TABELLE 3–1 Beschreibungen der Eigenschaften für ZFS-Pools

Eigenschaft	Typ	Standardwert	Beschreibung
`allocated`	Zeichenkette	entf.	Schreibgeschützter Wert, der die Menge des belegten Speicherplatzes im Pool angibt, der physisch zugewiesen ist.
`altroot`	Zeichenkette	`off`	Das alternative Root-Verzeichnis. Ist diese Eigenschaft gesetzt, wird dieses Verzeichnis jedem Einhängepunkt innerhalb des Pools vorangestellt. Diese Eigenschaft ist nützlich, um einen unbekannten Pool zu untersuchen, wenn die Einhängepunkte nicht vertrauenswürdig sind oder wenn in einer alternativen Boot-Umgebung die üblichen Pfade nicht gültig sind.
`autoreplace`	Boolesch	`off`	Regelt den automatischen Austausch von Speichergeräten. Wenn diese Eigenschaft auf `off` gesetzt ist, muss das Auswechseln von Speichergeräten mithilfe des Befehls zpool replace `eingeleitet werden.` Wenn diese Eigenschaft auf `on` gesetzt ist, wird das neue Speichergerät an der physischen Adresse des vorherigen Speichergeräts im Pool automatisch formatiert und in den Pool eingebunden. Die Abkürzung der Eigenschaft lautet `replace`.
`bootfs`	Boolesch	entf.	Das bootfähige Standarddateisystem für den Root-Pool. Diese Eigenschaft wird in der Regel vom Installationsprogramm gesetzt.
`cachefile`	Zeichenkette	entf.	Mit dieser Eigenschaft wird bestimmt, wo Pool-Konfigurationsinformationen im Cache gespeichert werden. Alle Pools im Cache werden beim Booten des Systems automatisch importiert. Installations- und Cluster-Umgebungen können jedoch erfordern, dass diese Informationen an anderer Stelle im Cache gespeichert werden, sodass Pools nicht automatisch importiert werden. Sie können diese Eigenschaft so einstellen, dass Poolkonfigurationen an einer anderen Stelle im Cache-Speicher abgelegt werden. Diese Informationen können später mithilfe des Befehls `zpool import -c` importiert werden. Bei den meisten ZFS-Konfigurationen wird diese Eigenschaft nicht verwendet.
`capacity`	Zahl	entf.	Schreibgeschützter Wert, der die Menge des belegten Speicherplatzes im Pool als Verhältnis zur Gesamtkapazität in Prozent angibt. Die Abkürzung der Eigenschaft lautet `cap`.

TABELLE 3–1 Beschreibungen der Eigenschaften für ZFS-Pools *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`delegation`	Boolesch	`on`	Bestimmt, ob einem Benutzer ohne ausreichende Berechtigungen die für das Dateisystem festgelegten Zugriffsrechte erteilt werden können. Weitere Informationen finden Sie in Kapitel 8, „Delegierte Oracle Solaris ZFS-Administration“.
`failmode`	Zeichenkette	`wait`	Steuert das Systemverhalten, wenn ein schwerwiegender Pool-Ausfall auftritt. Dieser Umstand ist in der Regel das Ergebnis des Konnektivitätsverlusts eines oder mehrerer zugrunde liegender Speichergeräte oder eines Ausfalls sämtlicher Geräte im Pool. In solch einem Fall wird das Verhalten durch einen der folgenden Werte bestimmt: ■ `wait` – Blockiert sämtliche E/A-Anforderungen, bis die Gerätekonnektivität wiederhergestellt ist und die Fehler mit dem Befehl `zpool clear` zurückgesetzt wurden. In diesem Zustand werden die E/A-Vorgänge, die den Pool betreffen, blockiert. Lesevorgänge können jedoch eventuell ausgeführt werden. Ein Pool bleibt im Wartezustand, bis das Geräteproblem behoben ist. ■ `continue` – Gibt bei jeder neuen E/A-Schreibanforderung einen E/A-Fehler zurück, lässt aber Lesezugriffe auf die übrigen fehlerfreien Speichergeräte zu. Schreibanforderungen, die noch auf der Festplatte festgeschrieben werden müssen, werden blockiert. Nach Wiederherstellung der Verbindung oder Ersetzen des Geräts müssen die Fehler mit dem Befehl `zpool clear` zurückgesetzt werden. ■ `panic` – Gibt eine Meldung an die Konsole aus und generiert einen Systemabsturz-Speicherabzug.
`free`	Zeichenkette	entf.	Schreibgeschützter Wert, der die Menge von Datenblöcken im Pool angibt, die nicht zugewiesen sind.
`guid`	Zeichenkette	entf.	Schreibgeschützte Eigenschaft und eindeutige Kennzeichnung des Pools.
`health`	Zeichenkette	entf.	Schreibgeschützte Eigenschaft, die den aktuellen Zustand des Pools angibt. Dabei bestehen folgende Möglichkeiten: ONLINE, DEGRADED, SUSPENDED, REMOVED oder UNAVAIL
`listshares`	Zeichenkette	off	Kontrolliert, ob Share-Informationen in diesem Pool mit dem Befehl `zfs list` angezeigt werden. Der Standardwert ist `off`.

TABELLE 3–1 Beschreibungen der Eigenschaften für ZFS-Pools *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`listsnapshots`	Zeichenkette	`on`	Kontrolliert, ob Schnappschuss-Informationen, die mit diesem Pool in Verbindung stehen, mit dem Befehl `zfs list` angezeigt werden. Wenn diese Eigenschaft deaktiviert wird, können die Schnappschuss-Informationen mit dem Befehl `zfs list -t snapshot` angezeigt werden.
`size`	Zahl	entf.	Schreibgeschützte Eigenschaft, die die Gesamtkapazität des Speicher-Pools angibt.
`version`	Zahl	entf.	Die aktuelle Version des Pools. Zum Aktualisieren von Pools wird grundsätzlich die Methode mit dem Befehl `zpool upgrade` empfohlen. Diese Eigenschaft bietet sich dann an, wenn aus Gründen der Abwärtskompatibilität eine spezifische Version benötigt wird. Diese Eigenschaft kann auf jede Zahl zwischen 1 und der aktuellen Version laut Ausgabe des Befehls `zpool upgrade -v` gesetzt werden.

Abfragen des Status von ZFS-Speicher-Pools

Mithilfe des Befehls `zpool list` können Sie mit unterschiedlichen Methoden Informationen zum Pool-Status abrufen. Die verfügbaren Informationen unterteilen sich im Allgemeinen in drei Kategorien: grundlegende Informationen zur Auslastung, E/A-Statistiken und Informationen zum Funktionsstatus. In diesem Abschnitt werden alle drei Kategorien dieser Informationen zu Speicher-Pools behandelt.

- „Anzeigen von Informationen zu ZFS-Speicher-Pools“ auf Seite 88
- „Anzeigen von E/A-Statistiken für ZFS-Speicher-Pools “ auf Seite 92
- „Ermitteln des Funktionsstatus von ZFS-Speicher-Pools“ auf Seite 94

Anzeigen von Informationen zu ZFS-Speicher-Pools

Mit dem Befehl `zpool list` können Sie grundlegende Pool-Informationen anzeigen.

Anzeigen von Informationen zu allen Speicherpools oder einem bestimmten Pool

Ohne Argumente werden mithilfe des Befehls `zpool list` folgende Informationen für alle Pools des Systems angezeigt:

```
# 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     -
```

Diese Befehlsausgabe zeigt folgende Informationen an:

`NAME`	Der Name des Pools.
`SIZE`	Die Gesamtkapazität des Pools entspricht der Summe der Speicherkapazität aller virtuellen Geräte der obersten Hierarchieebene.
`ALLOC`	Der von allen Datasets und internen Metadaten belegte physische Speicherplatz. Bitte beachten Sie, dass sich dieser Wert von der auf Dateisystemebene gemeldeten Festplattenkapazität unterscheidet. Weitere Informationen zum Ermitteln des verfügbaren Dateisystemspeicherplatzes finden Sie unter „Berechnung von ZFS-Festplattenkapazität“ auf Seite 32.
`FREE`	Der Wert des nicht belegten Speicherplatzes im Pool.
`CAP` (`CAPACITY`)	Der Wert der belegten Festplattenkapazität als Verhältnis zur Gesamtkapazität, in Prozent.
`HEALTH`	Der gegenwärtige Funktionsstatus des Pools. Weitere Informationen zum Pool-Status finden Sie unter „Ermitteln des Funktionsstatus von ZFS-Speicher-Pools“ auf Seite 94.
`ALTROOT`	Das alternative Root-Verzeichnis des Pools (falls vorhanden). Weitere Informationen zu Speicher-Pools mit alternativem Root-Verzeichnis finden Sie unter „Verwenden von ZFS-Speicher-Pools mit alternativem Root-Verzeichnis“ auf Seite 293.

Durch Angabe eines Pool-Namens können Sie sich auch Statistiken zu diesem bestimmten Pool anzeigen lassen. Beispiel:

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

Mithilfe der Zeitintervall- und Zählparameteroptionen von `zpool list` können Sie sich Statistiken über einen bestimmten Zeitraum anzeigen lassen. Außerdem können Sie mit der Option `-T` einen Zeitstempel anzeigen. Beispiel:

```
# zpool list -T d 3 2
Tue Nov  2 10:36:11 MDT 2010
NAME    SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
pool   33.8G  83.5K  33.7G     0%  1.00x  ONLINE  -
rpool  33.8G  12.2G  21.5G    36%  1.00x  ONLINE  -
Tue Nov  2 10:36:14 MDT 2010
pool   33.8G  83.5K  33.7G     0%  1.00x  ONLINE  -
rpool  33.8G  12.2G  21.5G    36%  1.00x  ONLINE  -
```

Anzeigen spezifischer Speicherpool-Statistikinformationen

Mithilfe der Option `-o` können Sie sich spezifische Statistikinformationen anzeigen lassen. Auch können Sie mithilfe dieser Option benutzerdefinierte Berichte erstellen oder sich gewünschte Informationen anzeigen lassen. Mit der folgenden Syntax wird beispielsweise nur der Name und die Speicherkapazität jedes Pools angezeigt:

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

Die Spaltentitel werden unter „Anzeigen von Informationen zu allen Speicherpools oder einem bestimmten Pool“ auf Seite 88 erläutert.

Verwenden von Ausgaben von ZFS-Speicher-Pools für Skripten

Die Standardausgabe des Befehls `zpool list` dient der Lesbarkeit am Bildschirm und ist für Shell-Skripten nicht zu verwenden. Zur Unterstützung programmatischer Verwendungen dieses Befehls kann mithilfe der Option `-H` die Ausgabe der Spaltentitel unterdrückt werden, und die einzelnen Felder werden durch Leerzeichen statt durch Tabulatoren getrennt. So rufen Sie beispielsweise eine Liste aller Pool-Namen im System mithilfe der folgenden Syntax ab:

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

Hier ist ein weiteres Beispiel:

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

Anzeige des Befehlsprotokolls von ZFS-Speicher-Pools

ZFS protokolliert automatisch `zfs`- und `zpool`-Befehle, durch die Pool-Zustandsinformationen geändert werden. Diese Informationen können mit dem Befehl `zpool history` angezeigt werden.

Die folgende Syntax zeigt beispielsweise die Befehlsausgabe für den Root-Pool:

```
# 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
```

Sie können auf Ihrem System eine ähnliche Ausgabe verwenden, um die ZFS-Befehle zu identifizieren, die bei der Behebung eines Fehlers ausgeführt wurden.

Das Verlaufsprotokoll zeichnet sich durch die folgenden Merkmale aus:

- Das Protokoll kann nicht deaktiviert werden.
- Das Protokoll wird persistent gespeichert, es wird also neustartübergreifend geführt.
- Das Protokoll wird in Form eines Ringpuffers implementiert. Die Mindestgröße beträgt 128 KB. Die Maximalgröße beträgt 32 MB.
- Für kleinere Pools ist die maximale Größe auf 1 Prozent der Pool-Größe beschränkt, wobei die *Größe* zum Zeitpunkt der Pool-Erstellung bestimmt wird.
- Das Protokoll erfordert keine Administration, d. h., eine Anpassung der Protokollgröße bzw. eine Änderung des Speicherorts sind nicht nötig.

Verwenden Sie zur Identifizierung des Befehlsprotokolls eines bestimten Speicher-Pools in etwa folgende Syntax:

```
# zpool history tank
2012-01-25.16:35:32 zpool create -f tank mirror c3t1d0 c3t2d0 spare c3t3d0
2012-02-17.13:04:10 zfs create tank/test
2012-02-17.13:05:01 zfs snapshot -r tank/test@snap1
```

Verwenden Sie die Option `-l` zum Anzeigen eines langen Formats mit Benutzernamen, Hostnamen und Angabe der Zone, in der der Vorgang ausgeführt wurde. Beispiel:

```
# zpool history -l tank
History for 'tank':
2012-01-25.16:35:32 zpool create -f tank mirror c3t1d0 c3t2d0 spare c3t3d0
[user root on tardis:global]
2012-02-17.13:04:10 zfs create tank/test [user root on tardis:global]
2012-02-17.13:05:01 zfs snapshot -r tank/test@snap1 [user root on tardis:global]
```

Verwenden Sie die Option `-i` zum Anzeigen interner Ereignisinformationen, die bei der Diagnose behilflich sein können. Beispiel:

```
# zpool history -i tank
History for 'tank':
2012-01-25.16:35:32 zpool create -f tank mirror c3t1d0 c3t2d0 spare c3t3d0
2012-01-25.16:35:32 [internal pool create txg:5] pool spa 33; zfs spa 33; zpl 5;
uts tardis 5.11 11.1 sun4v
2012-02-17.13:04:10 zfs create tank/test
```

```
2012-02-17.13:04:10 [internal property set txg:66094] $share2=2 dataset = 34
2012-02-17.13:04:31 [internal snapshot txg:66095] dataset = 56
2012-02-17.13:05:01 zfs snapshot -r tank/test@snap1
2012-02-17.13:08:00 [internal user hold txg:66102] <.send-4736-1> temp = 1 ...
```

Anzeigen von E/A-Statistiken für ZFS-Speicher-Pools

Mit dem Befehl `zpool iostat` können Sie E/A-Statistikinformationen für einen Pool bzw. ein virtuelles Datenspeichergerät abrufen. Dieser Befehl zeigt ähnlich wie der Befehl `iostat` eine statische "Momentaufnahme" aller E/A-Aktivitäten sowie aktualisierte Statistikinformationen für jedes definierte Zeitintervall an. Es werden die folgenden Statistikinformationen ausgegeben:

`alloc capacity` — Die Kapazität der gegenwärtig im Pool bzw. Gerät gespeicherten Daten. Aufgrund interner Implementierungsaspekte unterscheidet sich dieser Wert geringfügig von der für die betreffenden Dateisysteme verfügbaren Festplattenkapazität.

Weitere Informationen zu Unterschieden zwischen Pool- und Dataset-Speicherplatz finden Sie in „Berechnung von ZFS-Festplattenkapazität“ auf Seite 32.

`free capacity` — Die im Pool bzw. Gerät verfügbare Festplattenkapazität. Wie in der `used`-Statistik unterscheidet sich dieser Wert geringfügig von der für Datasets verfügbaren Festplattenkapazität.

`read operations` — Die Anzahl der zum Pool bzw. Gerät gesendeten E/A-Vorgänge einschließlich Metadaten-Anforderungen.

`write operations` — Die Anzahl der zum Pool bzw. Gerät gesendeten E/A-Schreiboperationen.

`read bandwidth` — Die Bandbreite aller Leseoperationen (einschließlich Metadaten) in Einheiten pro Sekunde.

`write bandwidth` — Die Bandbreite aller Schreiboperationen in Einheiten pro Sekunde.

Anzeigen globaler Pool-E/A-Statistikinformationen

Ohne Optionen zeigt der Befehl `zpool iostat` die bisher aufgelaufenen Statistikinformationen für alle Pools im System seit dem Hochfahren des Systems an. Beispiel:

```
# 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
```

```
----------  -----  -----  -----  -----  -----  -----
```

Da die angezeigten Statistikinformationen seit dem Hochfahren des Systems aufgelaufen sind, kann es sein, dass die Bandbreite bei relativ geringer Auslastung des Pools gering erscheint. Durch Angabe eines Zeitintervalls erhalten Sie ein realistischeres Bild der aktuellen Bandbreite. Beispiel:

```
# 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
```

Im obigen Beispiel zeigt der Befehl Auslastungsstatistiken für den Pool `tank` in Abständen von zwei Sekunden an, bis Sie Strg-C drücken. Alternativ können Sie ein zusätzliches `count`-Argument angeben, das den Befehl nach der angegebenen Anzahl von Iterationen beendet.

So gibt `zpool iostat 2 3` beispielsweise dreimal alle zwei Sekunden (also insgesamt sechs Sekunden lang) eine Übersicht aus. Wenn nur ein einziger Pool vorhanden ist, werden die Statistikinformationen in aufeinander folgenden Zeilen angezeigt. Sind mehrere Pools vorhanden, werden die Werte für die einzelnen Pools zur besseren Lesbarkeit durch gestrichelte Linien getrennt.

Anzeigen von E/A-Statistikinformationen zu virtuellen Geräten

Neben den globalen E/A-Statistikinformationen für einen Pool kann der Befehl `zpool iostat` außerdem E/A-Statistikinformationen für bestimmte virtuelle Datenspeichergeräte anzeigen. Mit diesem Befehl können Sie unverhältnismäßig langsame Datenspeichergeräte identifizieren oder die Verteilung der von ZFS generierten E/A-Vorgänge überprüfen. Zum Abrufen der vollständigen virtuellen Gerätestruktur sowie aller E/A-Statistikinformationen können Sie den Befehl `zpool iostat -v` nutzen. Beispiel:

```
# 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
----------  -----  -----  -----  -----  -----  -----
```

Bitte beachten Sie zwei wichtige Aspekte, wenn Sie E/A-Statistikinformationen für virtuelle Geräte abrufen:

- Erstens sind Statistikinformationen zur Belegung von Festplattenkapazität nur für virtuelle Geräte der obersten Hierarchieebene verfügbar. Die Art und Weise der Zuweisung von Festplattenkapazität bei virtuellen Geräten mit Datenspiegelung und RAID-Z ist implementierungsspezifisch und kann nicht in Form einer einzelnen Zahl ausgedrückt werden.
- Zweitens kann es sein, dass die einzelnen Werte nicht die erwartete Summe ergeben. Insbesondere sind Werte bei Geräten mit Datenspiegelung und RAID-Z nicht genau gleich. Diese Unterschiede sind besonders nach der Erstellung eines Pools bemerkbar, da ein großer Teil der E/A-Vorgänge infolge der Pool-Erstellung direkt auf den Datenträgern ausgeführt wird, was auf der Datenspiegelungsebene nicht berücksichtigt wird. Im Laufe der Zeit gleichen sich diese Werte allmählich an. Allerdings können auch defekte, nicht reagierende bzw. außer Betrieb genommen Geräte diese Symmetrie beeinträchtigen.

Bei der Untersuchung von Statistikinformationen zu virtuellen Geräten können Sie die gleichen Optionen (Zeitintervall und Zählparameter) verwenden.

Ermitteln des Funktionsstatus von ZFS-Speicher-Pools

ZFS bietet eine integrierte Methode zur Untersuchung der ordnungsgemäßen Funktion von Pools und Datenspeichergeräten. Der Funktionsstatus eines Pools wird aus dem Funktionsstatus aller seiner Datenspeichergeräte ermittelt. Diese Statusinformationen werden mit dem Befehl `zpool status` angezeigt. Außerdem werden potenzielle Pool- und Geräteausfälle von `fmd` gemeldet, an der Systemkonsole angezeigt und in der Datei `/var/adm/messages` protokolliert.

In diesem Abschnitt wird die Ermittlung des Funktionsstatus von Pools und Datenspeichergeräten erläutert. Dieses Kapitel enthält jedoch keine Informationen zur Reparatur fehlerhafter Pools bzw. Wiederherstellen des Normalbetriebs eines Pools. Weitere Informationen zur Fehlerbehebung und Datenwiederherstellung finden Sie in Kapitel 10, „Problembehebung und Pool-Wiederherstellung in Oracle Solaris ZFS“.

Der Integritätsstatus eines Pools wird mit einem von vier Status beschrieben:

`DEGRADED`
: Ein Pool mit mindestens einem fehlerhaften Gerät, die Daten sind jedoch wegen einer redundanten Konfiguration noch verfügbar.

`ONLINE`
: Alle Geräte eines Pools arbeiten normal.

`SUSPENDED`
: Ein Pool wartet darauf, dass die Verbindung zu einem Gerät wiederhergestellt wird. Ein `SUSPENDED` Pool bleibt im Wartezustand, bis das Geräteproblem behoben ist.

`UNAVAIL`
: Ein Pool mit beschädigten Metadaten, mit mindestens einem nicht verfügbaren Gerät und nicht ausreichenden Replikaten, um die Funktion fortsetzen zu können.

Poolgeräte können sich in einem der folgenden Zustände befinden:

`DEGRADED` Am virtuellen Gerät ist ein Fehler aufgetreten, es funktioniert jedoch noch. Dieser Zustand tritt am Häufigsten auf, wenn in einer RAID-Z-Konfiguration ein oder mehrere Datenspeichergeräte nicht mehr verfügbar sind. Die Fehlertoleranz des Pools kann beeinträchtigt werden, da ein Ausfall eines weiteren Geräts nicht behebbar sein könnte.

`OFFLINE` Das Gerät wurde vom Administrator außer Betrieb genommen.

`ONLINE` Das Gerät bzw. virtuelle Gerät arbeitet normal. Obwohl zeitweilige Übergangsfehler auftreten können, arbeitet das Gerät sonst einwandfrei.

`REMOVED` Das Gerät wurde bei laufendem Systembetrieb physisch ausgebaut. Die Erkennung ausgebauter Geräte ist von der jeweiligen Hardware abhängig und wird möglicherweise nicht auf allen Plattformen unterstützt.

`UNAVAIL` Mit dem Gerät bzw. virtuellen Gerät kann nicht kommuniziert werden. In manchen Fällen gehen Pools mit Geräten im Status `UNAVAIL` in den Status `DEGRADED`. Wenn sich ein virtuelles Gerät der obersten Hierarchieebene im Status `UNAVAIL` befindet, können von diesem Pool keine Daten abgerufen werden.

Der Funktionsstatus eines Pools wird aus dem Funktionsstatus aller seiner Datenspeichergeräte der obersten Hierarchieebene ermittelt. Befinden sich alle virtuellen Geräte eines Pools im Status `ONLINE`, besitzt der Pool ebenfalls den Status `ONLINE`. Befindet sich eines der virtuellen Geräte eines Pools im Status `DEGRADED` bzw. `UNAVAIL`, besitzt der Pool den Status `DEGRADED`. Wenn sich ein virtuelles Gerät der obersten Hierarchieebene im Status `UNAVAIL` bzw. `OFFLINE` befindet, hat der Pool ebenfalls den Status `UNAVAIL` oder `SUSPENDED`. Auf einen Pool im Status `UNAVAIL` oder `SUSPENDED` kann überhaupt nicht zugegriffen werden. Es können erst wieder Daten abgerufen werden, wenn erforderliche Datenspeichergeräte verbunden bzw. repariert werden. Ein Pool im Status `DEGRADED` bleibt zwar weiterhin aktiv, es kann jedoch sein, dass nicht das gleiche Datenredundanz- bzw. Datendurchsatzniveau wie bei einem ordnungsgemäßen Funktionieren des Pools erreicht wird.

Der Befehl `zpool status` stellt außerdem Details zu den Wiederherstellungs- und Bereinigungsvorgängen bereit.

- Bericht, während die Spiegelung läuft. Beispiel:

```
scan: resilver in progress since Wed Jun 20 14:19:38 2012
    7.43G scanned out of 71.8G at 36.4M/s, 0h30m to go
    7.43G resilvered, 10.35% done
```

- Bericht, während die Bereinigung läuft. Beispiel:

```
scan: scrub in progress since Wed Jun 20 14:56:52 2012
    529M scanned out of 71.8G at 48.1M/s, 0h25m to go
    0 repaired, 0.72% done
```

- Meldung, dass die Spiegelung abgeschlossen ist. Beispiel:

```
scan: resilvered 71.8G in 0h14m with 0 errors on Wed Jun 20 14:33:42 2012
```

- Meldung, dass die Bereinigung abgeschlossen ist. Beispiel:

```
scan: scrub repaired 0 in 0h11m with 0 errors on Wed Jun 20 15:08:23 2012
```

- Meldung, dass die laufende Bereinigung abgebrochen wurde. Beispiel:

```
scan: scrub canceled on Wed Jun 20 16:04:40 2012
```

- Meldungen zum Abschluss von Bereinigung und Spiegelung bleiben auch nach Systemneustarts erhalten.

Grundlegender Funktionsstatus eines Speicher-Pools

Sie können den Funktionsstatus eines Pools mithilfe des Befehls `zpool status` wie folgt rasch abrufen:

```
# zpool status -x
all pools are healthy
```

Bestimmte Pools können geprüft werden, indem der Pool-Name in der Befehlssyntax angegeben wird. Pools, die sich nicht im Status `ONLINE` befinden, sollten auf potenzielle Probleme untersucht werden (siehe folgender Abschnitt).

Ausführliche Informationen zum Funktionsstatus

Mithilfe der Option `-v` können Sie ausführlichere Informationen zum Funktionsstatus abrufen. Beispiel:

```
# 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
```

Diese Ausgabe enthält eine vollständige Beschreibung darüber, warum sich der Pool in seinem gegenwärtigen Funktionsstatus befindet. Es findet sich auch eine lesbare Erläuterung des Problems und ein Verweis auf einen Artikel in der Sun Knowledge Base, wenn Sie weitere Informationen dazu benötigen. Die Artikel der Sun Knowledge Base enthalten die aktuellsten Informationen zur Behebung eines bestimmten Problems. Mithilfe der aufgeführten ausführlichen Konfigurationsinformationen sollten Sie feststellen können, welches Datenspeichergerät defekt ist und wie Sie den Pool reparieren können.

Im vorherigen Beispiel muss das `UNAVAIL`-Datenspeichergerät ausgetauscht werden. Nach dem Austauschen des Geräts können Sie es falls erforderlich mit dem Befehl `zpool online` wieder in Betrieb nehmen. Beispiel:

```
# zpool online tank c1t0d0
Bringing device c1t0d0 online
# zpool status -x
all pools are healthy

# zpool online pond c0t5000C500335F907Fd0
warning: device 'c0t5000C500335DC60Fd0' onlined, but remains in degraded state
# zpool status -x
all pools are healthy
```

Mit der obigen Ausgabe wird angegeben, dass das Gerät in einem herabgestuften Status bleibt, bis die Wiederherstellung abgeschlossen ist.

Wenn die Eigenschaft `autoreplace` aktiviert ist, müssen Sie das ausgetauschte Gerät möglicherweise nicht in Betrieb nehmen.

Wenn in einem Pool ein außer Betrieb genommenes Gerät vorhanden ist, kann es mithilfe der vom Befehl ausgegebenen Informationen identifiziert werden. Beispiel:

```
# 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

# zpool status -x
 pool: pond
 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'.
config:

        NAME                      STATE     READ WRITE CKSUM
        pond                      DEGRADED     0     0     0
          mirror-0                DEGRADED     0     0     0
            c0t5000C500335F95E3d0 ONLINE       0     0     0
            c0t5000C500335F907Fd0 OFFLINE      0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000C500335BD117d0 ONLINE       0     0     0
            c0t5000C500335DC60Fd0 ONLINE       0     0     0

errors: No known data errors
```

In den Spalten `READ` und `WRITE` wird die Anzahl der für das betreffende Gerät gezählten E/A-Fehler angezeigt, und die Spalte `CKSUM` enthält die Anzahl der am Gerät aufgetretenen nicht behebbaren Prüfsummenfehler. Da beide Fehlerzähler auf einen wahrscheinlichen Geräteausfall hinweisen, sollten Sie Maßnahmen zur Behebung dieses Problems einleiten. Wenn für ein virtuelles Gerät der obersten Hierarchieebene Fehlerwerte angezeigt werden, die nicht gleich null sind, kann es sein, dass auf Daten teilweise nicht mehr zugegriffen werden kann

Das Feld `errors:` weist auf bekannte Datenfehler hin.

In der Befehlsausgabe des vorherigen Beispiels verursacht das außer Betrieb genommene Gerät keine Datenfehler.

Weitere Informationen zum Auffinden von Fehlern und Reparieren von `UNAVAIL` Pools und Daten finden Sie in Kapitel 10, „Problembehebung und Pool-Wiederherstellung in Oracle Solaris ZFS“.

Sammeln von Informationen des ZFS-Speicher-Pools

Mithilfe der Zeitintervall- und Zählparameteroptionen von `zpool status` können Sie sich Statistiken über einen bestimmten Zeitraum anzeigen lassen. Außerdem können Sie mit der Option `-T` einen Zeitstempel anzeigen. Beispiel:

```
# zpool status -T d 3 2
Wed Jun 20 16:10:09 MDT 2012
  pool: pond
 state: ONLINE
```

```
  scan: resilvered 9.50K in 0h0m with 0 errors on Wed Jun 20 16:07:34 2012
config:

        NAME                      STATE     READ WRITE CKSUM
        pond                      ONLINE       0     0     0
          mirror-0                ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0

errors: No known data errors

  pool: rpool
 state: ONLINE
  scan: scrub repaired 0 in 0h11m with 0 errors on Wed Jun 20 15:08:23 2012
config:

        NAME                        STATE     READ WRITE CKSUM
        rpool                       ONLINE       0     0     0
          mirror-0                  ONLINE       0     0     0
            c0t5000C500335BA8C3d0s0  ONLINE       0     0     0
            c0t5000C500335FC3E7d0s0  ONLINE       0     0     0

errors: No known data errors
Wed Jun 20 16:10:12 MDT 2012

  pool: pond
 state: ONLINE
  scan: resilvered 9.50K in 0h0m with 0 errors on Wed Jun 20 16:07:34 2012
config:

        NAME                      STATE     READ WRITE CKSUM
        pond                      ONLINE       0     0     0
          mirror-0                ONLINE       0     0     0
            c0t5000C500335F95E3d0  ONLINE       0     0     0
            c0t5000C500335F907Fd0  ONLINE       0     0     0
          mirror-1                ONLINE       0     0     0
            c0t5000C500335BD117d0  ONLINE       0     0     0
            c0t5000C500335DC60Fd0  ONLINE       0     0     0

errors: No known data errors

  pool: rpool
 state: ONLINE
  scan: scrub repaired 0 in 0h11m with 0 errors on Wed Jun 20 15:08:23 2012
config:

        NAME                        STATE     READ WRITE CKSUM
        rpool                       ONLINE       0     0     0
          mirror-0                  ONLINE       0     0     0
            c0t5000C500335BA8C3d0s0  ONLINE       0     0     0
            c0t5000C500335FC3E7d0s0  ONLINE       0     0     0

errors: No known data errors
```

Migrieren von ZFS-Speicher-Pools

Gelegentlich kann es vorkommen, dass Speicher-Pools zwischen Systemen transferiert werden müssen. Dafür müssen Sie die Datenspeichergeräte aus dem ursprünglichen System herausnehmen und an das neue System anschließen. Dies kann durch Neuverkabelung der betreffenden Geräte bzw. die Verwendung von Geräten mit mehreren Anschlüssen, z. B. Geräte in einem Speichernetzwerk (SAN), bewerkstelligt werden. Mit ZFS können Sie einen Pool aus einem System exportieren und in das Zielsystem importieren. Dies ist auch möglich, wenn beide Systemarchitekturen unterschiedliche Bitbreiten besitzen. Informationen zum Replizieren bzw. Migrieren von Dateisystemen zwischen verschiedenen Speicherpools, die auf unterschiedlichen Systemen installiert sind, finden Sie unter „Senden und Empfangen von ZFS-Daten“ auf Seite 227.

Vorbereiten der Migration eines ZFS-Speicher-Pools

Speicher-Pools sollten explizit exportiert werden, um anzuzeigen, dass sie zur Migration bereit sind. Bei diesem Vorgang werden ungeschriebene Daten auf die Festplatte ausgespeichert und Daten auf die Festplatte geschrieben, wodurch angezeigt wird, dass der Export abgeschlossen ist. Anschließend werden alle Informationen, die den Pool betreffen, aus dem System entfernt.

Wenn Sie den Pool nicht explizit exportieren, sondern die Datenträger manuell entfernen, kann der resultierende Pool trotzdem noch in ein anderes System importiert werden. Es kann jedoch sein, dass die in den allerletzten Sekunden ausgeführten Datentransaktionen verloren gehen und der betreffende Pool auf dem ursprünglichen System als `UNAVAIL` angezeigt wird, da die Geräte nicht mehr vorhanden sind. Standardmäßig kann ein Pool, der nicht explizit exportiert wurde, nicht vom Zielsystem importiert werden. Dies ist erforderlich, um zu verhindern, dass versehentlich ein aktiver Pool importiert wird, der Speicherplatz enthält, der über das Netzwerk zugänglich ist und noch von einem anderem System belegt wird.

Exportieren eines ZFS-Speicher-Pools

Speicher-Pools können mit dem Befehl `zpool export` exportiert werden. Beispiel:

```
# zpool export tank
```

Vor dem Fortfahren versucht der Befehl alle innerhalb des Pools eingehängten Dateisysteme auszuhängen. Falls das Aushängen von Dateisystemen fehlschlägt, können Sie mithilfe der Option `-f` ein Aushängen erzwingen. Beispiel:

```
# zpool export tank
cannot unmount '/export/home/eric': Device busy
# zpool export -f tank
```

Nach der Ausführung dieses Befehls ist der Pool `tank` im System nicht mehr sichtbar.

Falls Datenspeichergeräte zum Zeitpunkt des Exports nicht verfügbar sind, können die betreffenden Geräte nicht als "sauber" exportiert eingestuft werden. Wenn ein solches Datenspeichergerät später ohne die funktionierenden Datenspeichergeräte mit einem System verbunden wird, erscheint das betreffende Gerät als potenziell aktiv.

Wenn im Pool ZFS-Volumes vorhanden sind, kann der Pool auch nicht mithilfe der Option `-f` exportiert werden. Wenn Sie einen Pool mit einem ZFS-Volume exportieren möchten, müssen Sie zunächst sicherstellen, dass das Volume nicht von aktiven Ressourcen belegt wird.

Weitere Informationen zu ZFS-Volumes finden Sie unter „ZFS-Volumes“ auf Seite 283.

Ermitteln verfügbarer Speicher-Pools für den Import

Nachdem ein Pool (durch expliziten Export oder erzwungenes Entfernen von Datenspeichergeräten) aus einem System entfernt wurde, müssen die betreffenden Geräte mit dem Zielsystem verbunden werden. ZFS kann Situationen bewältigen, in denen nur einige der Geräte verfügbar sind. Der Erfolg der Pool-Migration hängt jedoch von der Funktionstüchtigkeit der Geräte ab. Die Geräte müssen jedoch nicht notwendigerweise unter dem gleichen Gerätenamen verbunden werden. ZFS erkennt verschobene bzw. umbenannte Geräte und passt die Konfiguration entsprechend an. Führen Sie zum Ermitteln importierbarer Pools den Befehl `zpool export` aus. Beispiel:

```
# 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
```

In diesem Beispiel kann der Pool `tank` in ein Zielsystem importiert werden. Jeder Pool wird durch einen Namen und einen eindeutigen numerischen Bezeichner identifiziert. Wenn

mehrere Pools mit dem gleichen Namen für den Import verfügbar sind, können sie mithilfe des numerischen Bezeichners unterschieden werden.

Ebenso wie die Ausgabe des Befehls `zpool status` verweist die Ausgabe des Befehls `zpool import` auf einen Artikel der Sun Knowledge Base, der die aktuellsten Informationen und Reparaturhinweise zu Problemen enthält, die das Importieren von Pools verhindern. In diesem Fall kann das Importieren eines Pools erzwungen werden. Das Importieren eines Pools, der gegenwärtig über Netzwerkzugriff von einem anderen System verwendet wird, kann Daten beschädigen und auf beiden Systemen zu Abstürzen führen, wenn diese Systeme Daten auf das gleiche Datenspeichergerät schreiben. Wenn einige Geräte eines Pools nicht verfügbar sind, zum Bereitstellen eines funktionierenden Pools jedoch genügend Redundanzdaten vorhanden sind, geht der Pool in den Status `DEGRADED`. Beispiel:

```
# 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
```

In diesem Beispiel ist der erste Datenträger beschädigt oder nicht vorhanden, der Pool kann aber trotzdem importiert werden, da die gespiegelten Daten noch verfügbar sind. Wenn zu viele Datenspeichergeräte nicht verfügbar sind, kann der Pool nicht importiert werden.

In diesem Beispiel fehlen in einem virtuellen RAID-Z-Gerät zwei Datenträger, was bedeutet, dass zum Rekonstruieren des Pools nicht genügend Redundanz verfügbar ist. In einigen Fällen kann es auch sein, dass zum Ermitteln der vollständigen Konfiguration nicht genügend Datenspeichergeräte vorhanden sind. In einem solchen Fall kann ZFS nicht bestimmen, welche anderen Geräte zu diesem Pool gehört haben, obwohl ZFS diesbezüglich so viele Informationen wie möglich meldet. Beispiel:

```
# 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.
```

Importieren von ZFS-Speicher-Pools aus anderen Verzeichnissen

Standardmäßig durchsucht der Befehl `zpool import` nur im Verzeichnis `/dev/dsk` enthaltene Datenspeichergeräte. Wenn Geräte in einem anderen Verzeichnis vorhanden sind oder Sie Pools verwenden, die durch Dateien gesichert sind, müssen Sie mithilfe der Option `-d` andere Verzeichnisse durchsuchen. Beispiel:

```
# 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
```

Sie können die Option `-d` mehrmals angeben, wenn Datenspeichergeräte in mehreren Verzeichnissen vorhanden sind.

Importieren von ZFS-Speicher-Pools

Wenn ein Pool für den Import ermittelt wurde, können Sie ihn durch Angabe seines Namens oder numerischen Bezeichners als Argument für den Befehl `zpool import` importieren. Beispiel:

```
# zpool import tank
```

Wenn mehrere Pools den gleichen Namen besitzen, müssen Sie mithilfe des numerischen Bezeichners angeben, welcher Pool importiert werden soll. Beispiel:

```
# 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
```

Wenn ein Pool-Name mit einem bereits vorhandenen Pool-Namen in Konflikt steht, können Sie den Pool unter einem anderen Namen importieren. Beispiel:

```
# zpool import dozer zeepool
```

Dieser Befehl importiert den exportierten Pool `dozer` unter dem neuen Namen `zeepool`. Der Name des neuen Pools ist dauerhaft.

Wenn ein Pool nicht ordnungsgemäß exportiert wurde, benötigt ZFS das `-f`-Flag, um zu verhindern, dass versehentlich ein Pool importiert wird, der noch von einem anderen System benutzt wird. Beispiel:

```
# zpool import dozer
cannot import 'dozer': pool may be in use on another system
use '-f' to import anyway
# zpool import -f dozer
```

Hinweis – Versuchen Sie nicht, einen in einem System aktiven Pool in ein anderes System zu importieren. ZFS ist kein natives Cluster-, verteiltes oder paralleles Dateisystem und bietet keinen gleichzeitigen Zugriff von mehreren verschiedenen Hosts aus.

Pools können mithilfe der Option `-R` in ein anderes Root-Verzeichnis importiert werden. Weitere Informationen zu Speicher-Pools mit alternativem Root-Verzeichnis finden Sie unter „Verwenden von ZFS-Speicher-Pools mit alternativem Root-Verzeichnis“ auf Seite 293.

Importieren eines Pools mit fehlendem Protokolliergerät

Standardmäßig kann ein Pool mit einem fehlenden Protokolliergerät nicht importiert werden. Mit dem Befehl `zpool import -m` können Sie den Import eines Pools mit einem fehlenden Protokolliergerät erzwingen. Beispiel:

```
# zpool import dozer
The devices below are missing, use '-m' to import the pool anyway:
            c3t3d0 [log]

cannot import 'dozer': one or more devices is currently unavailable
```

Importieren Sie den Pool mit dem fehlenden Protokolliergerät. Beispiel:

```
# zpool import -m dozer
# zpool status dozer
  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
  scan: none requested
config:

        NAME                    STATE     READ WRITE CKSUM
        dozer                   DEGRADED     0     0     0
          mirror-0              ONLINE       0     0     0
            c8t0d0              ONLINE       0     0     0
            c8t1d0              ONLINE       0     0     0
        logs
          2189413556875979854   UNAVAIL      0     0     0

errors: No known data errors
```

Führen Sie nach dem Anschließen des fehlenden Protokolliergeräts den Befehl `zpool clear` aus, um die Pool-Fehler zu löschen.

Eine ähnliche Wiederherstellung kann mit fehlenden gespiegelten Protokolliergeräten versucht werden. Beispiel:

```
# zpool import dozer
The devices below are missing, use '-m' to import the pool anyway:
            mirror-1 [log]
              c3t3d0
              c3t4d0

cannot import 'dozer': one or more devices is currently unavailable
# zpool import -m dozer
# zpool status dozer
  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
 scan: scrub repaired 0 in 0h0m with 0 errors on Fri Oct 15 16:51:39 2010
config:

        NAME                      STATE     READ WRITE CKSUM
        dozer                     DEGRADED     0     0     0
          mirror-0                ONLINE       0     0     0
            c3t1d0                ONLINE       0     0     0
            c3t2d0                ONLINE       0     0     0
        logs
          mirror-1                UNAVAIL      0     0     0  insufficient replicas
            13514061426445294202  UNAVAIL      0     0     0  was c3t3d0
            16839344638582008929  UNAVAIL      0     0     0  was c3t4d0
```

Führen Sie nach dem Anschließen der fehlenden Protokolliergeräte den Befehl `zpool clear` aus, um die Pool-Fehler zu löschen.

Importieren eines Pools im schreibgeschützten Modus

Sie können einen Pool im schreibgeschützten Modus importieren. Ist ein Pool dermaßen beschädigt, dass nicht auf ihn zugegriffen werden kann, können Sie mithilfe dieser Funktion die Daten des Pools eventuell wiederherstellen. Beispiel:

```
# zpool import -o readonly=on tank
# zpool scrub tank
cannot scrub tank: pool is read-only
```

Beim Importieren eines Pools im schreibgeschützten Modus gelten die folgenden Bedingungen:

- Alle Dateisysteme und Volumes werden im schreibgeschützten Modus eingehängt.
- Die Pool-Transaktionsverarbeitung ist deaktiviert. Dies bedeutet auch, dass alle anstehenden synchronen Schreibvorgänge im Intent-Protokoll nicht wiedergegeben werden, ehe der Pool mit Lese- und Schreibzugriff importiert wird.
- Jegliche Versuche, während des Imports mit Lese- und Schreibzugriff eine Pool-Eigenschaft festzulegen, werden ignoriert.

Ein schreibgeschützter Pool kann durch Exportieren und Importieren in den Lese- und Schreibmodus zurückgesetzt werden. Beispiel:

```
# zpool export tank
# zpool import tank
# zpool scrub tank
```

Importieren eines Pools nach einem spezifischen Gerätepfad

Der folgende Befehl importiert den Pool `dpool`, indem er eines der spezifischen Geräte (hier `/dev/dsk/c2t3d0`) identifiziert.

```
# zpool import -d /dev/dsk/c2t3d0s0 dpool
# zpool status dpool
  pool: dpool
 state: ONLINE
 scan: resilvered 952K in 0h0m with 0 errors on Fri Jun 29 16:22:06 2012
config:

        NAME        STATE     READ WRITE CKSUM
        dpool       ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            c2t3d0  ONLINE       0     0     0
            c2t1d0  ONLINE       0     0     0
```

Der Befehl muss den Bereichsbezeichner des spezifischen Geräts einbeziehen, obwohl dieser Pool aus ganzen Festplatten besteht.

Wiederherstellen gelöschter ZFS-Speicher-Pools

Gelöschte Speicher-Pools können mit dem Befehl `zpool import -D` wiederhergestellt werden. Beispiel:

```
# 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
```

In dieser Ausgabe des Befehls `zpool import` kann der `tank`-Pool aufgrund der folgenden Statusinformationen als gelöscht erkannt werden:

```
state: ONLINE (DESTROYED)
```

Führen Sie zum Wiederherstellen des gelöschten Pools den Befehl `zpool import -D` erneut aus. Beispiel:

```
# 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
```

Wenn eines der Geräte im gelöschten Pool nicht verfügbar ist, können Sie den gelöschten Pool unter Umständen mit der Option `-f` dennoch wiederherstellen. Importieren Sie in einer solchen Situation den im eingeschränkten Zustand befindlichen Pool und versuchen Sie dann, den Geräteausfall zu beheben. Beispiel:

```
# zpool destroy dozer
# zpool import -D
  pool: dozer
    id: 4107023015970708695
 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:
        dozer              DEGRADED
          raidz2-0         DEGRADED
            c8t0d0         ONLINE
            c8t1d0         ONLINE
            c8t2d0         ONLINE
            c8t3d0         UNAVAIL  cannot open
            c8t4d0         ONLINE
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
   scan: none requested
config:

        NAME                     STATE     READ WRITE CKSUM
        dozer                    DEGRADED     0     0     0
          raidz2-0               DEGRADED     0     0     0
            c8t0d0               ONLINE       0     0     0
            c8t1d0               ONLINE       0     0     0
            c8t2d0               ONLINE       0     0     0
            4881130428504041127  UNAVAIL      0     0     0
            c8t4d0        ONLINE       0     0     0

errors: No known data errors
# zpool online dozer c8t4d0
# zpool status -x
all pools are healthy
```

Aktualisieren von ZFS-Speicher-Pools

Wenn in Ihrem System ZFS-Speicherpools aus einer früheren Solaris-Version vorhanden sind, können Sie diese Pools mit dem Befehl `zpool upgrade` aktualisieren, um die Poolfunktionen der aktuellen Version nutzen zu können. Darüber hinaus weist der Befehl `zpool status` Sie darauf hin, wenn Pools mit älteren Versionen laufen. Beispiel:

```
# 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
```

Mithilfe der folgenden Syntax können Sie zusätzliche Informationen zu einer bestimmten Version und unterstützten Releases ermitteln:

```
# 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.
```

Anschließend können Sie den Befehl `zpool upgrade` ausführen, um alle Pools zu aktualisieren. Beispiel:

```
# zpool upgrade -a
```

Hinweis – Wenn Sie den Pool auf eine neuere ZFS-Version aktualisieren, ist er auf einem System, auf dem eine ältere ZFS-Version ausgeführt wird, nicht verfügbar.

Wenn Sie die ZFS-Administrationskonsole auf einem System mit einem Pool aus früheren Solaris-Versionen nutzen möchten, müssen Sie die Pools vor Verwendung der ZFS-Administrationskonsole aktualisieren. Mit dem Befehl `zpool status` lässt sich feststellen, ob Pools aktualisiert werden müssen.

◆ ◆ ◆ KAPITEL 4

Installieren und Booten eines Oracle Solaris ZFS-Root-Dateisystems

In diesem Kapitel erfahren Sie, wie ein Oracle Solaris ZFS-Root-Dateisystem installiert und gebootet wird. Darüber hinaus wird die Migration eines UFS-Root-Dateisystems in ein ZFS-Dateisystem mithilfe von Oracle Solaris Live Upgrade behandelt.

Dieses Kapitel enthält die folgenden Abschnitte:

Eine Liste der bekannten Probleme in dieser Version finden Sie in den *Oracle Solaris 10 1/13 - Versionshinweise*.

Installieren und Booten eines Oracle Solaris ZFS-Root-Dateisystems (Übersicht)

Folgende Methoden zum Installieren und Booten eines ZFS-Root-Dateisystems stehen zur Verfügung:

- **Erstinstallation von Oracle Solaris (interaktives Textmodus-Installationsverfahren)**
  - Auswählen und Installieren von ZFS als Root-Dateisystem
  - Installieren eines ZFS-Flash-Archivs
- **Oracle Solaris Live Upgrade**
  - Migrieren eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem
  - Erstellen einer neuen Boot-Umgebung in einem neuen ZFS-Root-Pool.
  - Erstellen oder Aktualisieren einer Boot-Umgebung in einem vorhandenen ZFS-Root-Pool
  - Aktualisieren einer alternativen Boot-Umgebung (BU) mit einem ZFS-Flash-Archiv
- **Oracle Solaris JumpStart**
  - Erstellen eines Profils zur automatischen Installation eines Systems mit einem ZFS-Root-Dateisystem
  - Erstellen eines Profils zur automatischen Installation eines Systems mit einem ZFS-Flash-Archiv

Nach der Installation eines ZFS-Root-Dateisystems oder der Migration in ein ZFS-Root-Dateisystem auf einem SPARC- oder x86-System startet das System automatisch aus dem ZFS-Root-Dateisystem. Weitere Informationen zum Ändern des Boot-Verhaltens finden Sie unter „Booten aus einem ZFS-Root-Dateisystem“ auf Seite 162.

Leistungsmerkmale für die ZFS-Installation

In dieser Oracle Solaris-Version stehen die folgenden Leistungsmerkmale für die ZFS-Installation zur Verfügung:

- Das interaktive textbasierte Installationsprogramm ermöglicht die Installation eines UFS- oder ZFS-Root-Dateisystems. In dieser Version ist UFS weiterhin das Standarddateisystem. Auf das interaktive textbasierte Installationsprogramm können Sie wie folgt zugreifen:
  - SPARC: Verwenden Sie die folgende Syntax für die Oracle Solaris-Installations-DVD:

    ```
    ok boot cdrom - text
    ```
  - SPARC: Verwenden Sie die folgende Syntax beim Booten über das Netzwerk:

    ```
    ok boot net - text
    ```
  - x86: Wählen Sie das Textmodus-Installationsverfahren.

- Ein benutzerdefiniertes JumpStart-Profil bietet folgende Funktionen:
  - Sie können ein Profil zum Erstellen eines ZFS-Speicher-Pools anlegen und ein bootfähiges ZFS-Dateisystem benennen.
  - Sie können ein Profil zur Installation eines Flash-Archivs eines ZFS-Root-Pools anlegen.
- Sie können ein UFS-Root-Dateisystem mithilfe des Live Upgrade in ein ZFS-Root-Dateisystem migrieren.
- Sie können bei der Installation zwei Festplatten auswählen und einen gespiegelten Speicher-Pool mit ZFS-Root-Dateisystem (im Folgenden "ZFS-Root-Pool") einrichten. Alternativ lassen sich auch nach der Installation zusätzliche Festplatten anhängen, um einen gespiegelten ZFS-Root-Pool herzustellen.
- Swap- und Dump-Geräte werden automatisch auf ZFS-Volumes im ZFS-Root-Pool erstellt.

Die folgenden Installationsfunktionen stehen in diesem Release nicht zur Verfügung:

- Die GUI-Installationsfunktion steht derzeit nicht zum Installieren eines ZFS-Root-Dateisystems zur Verfügung. Sie müssen das Textmodus-Installationsverfahren auswählen, um ein ZFS-Root-Dateisystem zu installieren.
- Ein Upgrade des UFS-Root-Dateisystems zu einem ZFS-Root-Dateisystem ist nicht mithilfe des Standard-Upgrade-Programms möglich.

Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung

Vergewissern Sie sich vor dem Versuch, ein ZFS-Root-Dateisystem auf einem System zu installieren oder ein UFS- in ein ZFS-Root-Dateisystem zu migrieren, dass folgende Voraussetzungen erfüllt sind:

Voraussetzungen für die Oracle Solaris-Version

Sie haben folgende Möglichkeiten, ein ZFS-Root-Dateisystem zu installieren und zu booten oder in ein ZFS-Root-Dateisystem zu migrieren:

- Installieren eines ZFS-Root-Dateisystems – ab Solaris 10 10/08 verfügbar.
- Migrieren aus einem UFS-Root-Dateisystem in ein ZFS-Root-Dateisystem mit Live Upgrade – Sie müssen Solaris 10 10/08 oder eine höhere Version installiert oder auf Solaris 10 10/08 oder eine höhere Version aktualisiert haben.

Allgemeine Anforderungen an den ZFS-Root-Pool

In den folgenden Abschnitte werden der Speicherplatz für ZFS-Root-Pools und Konfigurationsvoraussetzungen beschrieben.

Erforderliche Festplattenkapazität für ZFS-Root-Pools

Ein ZFS-Root-Dateisystem benötigt mehr Speicherplatz im Pool als ein UFS-Root-Dateisystem, da Swap- und Dump-Geräte in einer ZFS-Root-Umgebung separate Speichergeräte sein müssen. In UFS-Root-Dateisystemen sind Swap- und Dump-Geräte standardmäßig dasselbe Gerät.

Wenn auf einem System ein ZFS-Root-Dateisystem installiert oder ein Upgrade darauf vorgenommen wird, hängen die Größe des Swap-Bereichs und des Dump-Geräts von der physischen Speicherkapazität ab. Die Mindestgröße des verfügbaren Pool-Speicherplatzes im Pool für ein bootfähiges ZFS-Root-Dateisystem richtet sich nach der Größe des physischen Speichers, dem freien Speicherplatz auf der Festplatte und der Anzahl der zu erstellenden Boot-Umgebungen (BUs).

Beachten Sie die erforderliche Festplattenkapazität für ZFS-Speicher-Pools:

- Für die Installation eines ZFS-Root-Dateisystems sind mindestens 1536 MB Arbeitsspeicher erforderlich.
- Für die optimale Leistung des gesamten ZFS-Dateisystems werden mindestens 1536 MB empfohlen.
- Empfohlen werden mindestens 16 GB freier Speicherplatz auf der Festplatte. Der Festplattenspeicher wird wie folgt belegt:
  - **Swap-Bereich und Dump-Gerät** – Die Solaris-Installationsprogramme erstellen Swap- und Dump-Volumes in den folgenden Standardgrößen:
    - **Erstinstallation** – Die Standard-Swap-Größe ist in der neuen ZFS-Boot-Umgebung halb so groß wie der physische Speicher, im Allgemeinen zwischen 512 MB und 2 GB. Die Größe des Swap-Volume kann während einer Erstinstallation angepasst werden.
    - Die Standard-Dump-Größe wird vom Kernel basierend auf den `dumpadm`-Informationen und der Größe des physischen Speichers berechnet. Die Größe des Dump-Volume kann während einer Erstinstallation angepasst werden.
    - **Live Upgrade** – Wenn ein UFS-Root-Dateisystem auf ein ZFS-Root-Dateisystem umgestellt wird, wird die Standard-Swap-Größe für die ZFS-Boot-Umgebung (ZFS-BU) als Größe des Swap-Geräts der UFS-BU berechnet. Bei der Berechnung der Standard-Swap-Größe werden die Größen aller Swap-Geräte in der UFS-BU addiert, und es wird ein ZFS-Volume der entsprechenden Größe in der ZFS-BU erstellt. Wenn keine Swap-Geräte in der UFS-BU definiert sind, wird die Standard-Swap-Größe auf 512 MB gesetzt.
    - Die Standard-Dump-Größe ist in der ZFS-BU halb so groß wie der physische Speicher, zwischen 512 MB und 2 GB.

    Sie können die Größen der Swap- und Dump-Volumes ändern, sofern die neuen Größen den Betrieb des Systems unterstützen. Weitere Informationen finden Sie unter „Anpassen der Größe von ZFS-Swap- und Dump-Geräten“ auf Seite 159.

- **Boot-Umgebung (BU)** – Zusätzlich zu dem für neue Swap- und Dump-Geräte erforderlichen oder nachträglich geänderten Speicherplatz werden für eine ZFS-BU bei Migration von einer UFS-BU ungefähr 6 GB benötigt. Für ZFS-BUs, die aus anderen ZFS-BUs geklont werden, ist kein zusätzlicher Speicherplatz erforderlich. Beachten Sie jedoch, dass die BU-Größe durch die Installation von Patches zunimmt. Alle ZFS-BUs in demselben Root-Pool greifen auf dieselben Swap- und Dump-Geräte zu.
- **Komponenten des Betriebssystems Oracle Solaris** – Alle Unterverzeichnisse des Root-Dateisystems, die zum Betriebssystem-Image gehören, ausgenommen `/var`, müssen im selben Dataset wie das Root-Dateisystem enthalten sein. Außerdem müssen alle Komponenten des Betriebssystems im Root-Pool enthalten sein, mit Ausnahme der Swap- und Dump-Geräte.

  Informationen zur Änderung von Standard-Swap- und -Dump-Geräten mit Live Upgrade finden Sie in „Anpassen der ZFS-Swap- und Dump-Volumes“ auf Seite 161.

  Eine weitere Beschränkung ist, dass das Verzeichnis bzw. Dataset `/var` ein einzelnes Dataset sein muss. Sie können kein untergeordnetes `/var`-Dataset wie beispielsweise `/var/tmp` erstellen, wenn Sie Live Upgrade verwenden möchten, um eine ZFS-BU zu migrieren oder zu patchen oder ein ZFS-Flash-Archiv dieses Pools zu erstellen.

Beispielsweise könnte ein System mit 12 GB Festplattenkapazität zu klein für eine bootfähige ZFS-Umgebung sein, da 2 GB je Swap- und Dump-Gerät und rund 6 GB für die ZFS-BU benötigt werden, die aus einer UFS-BU migriert wird.

Anforderungen an die Konfiguration des ZFS-Root-Pools

Machen Sie sich mit folgenden Voraussetzungen für die Konfiguration von ZFS-Root-Pools vertraut:

- Der als Root-Pool bestimmte Pool muss ein SMI-Label haben. Diese Anforderung wird üblicherweise erfüllt, wenn der Pool mithilfe von Festplattenbereichen erstellt wird.
- Der Pool muss entweder auf einem Festplattenbereich oder auf gespiegelten Festplattenbereichen vorhanden sein. Bei dem Versuch, eine nicht unterstützte Pool-Konfiguration bei einer Live Upgrade-Migration zu verwenden, wird eine Meldung folgender Art angezeigt:

  ```
  ERROR: ZFS pool name does not support boot environments
  ```

  Weitere Informationen zu unterstützten Konfigurationen von ZFS-Root-Pools finden Sie unter „Erstellen eines ZFS-Root-Pools“ auf Seite 54.
- x86: Die Festplatte muss eine Oracle Solaris-`fdisk`-Partition enthalten. Diese `fdisk`-Partition wird bei der Installation eines x86-Systems automatisch erstellt. Weitere Informationen zu Solaris-`fdisk`-Partitionen finden Sie in „Guidelines for Creating an fdisk Partition“ in *System Administration Guide: Devices and File Systems*.
- Die Größe von Datenträgern, die für das Booten in einem ZFS-Root-Pool bestimmt sind, muss sowohl auf SPARC- als auch auf x86-Systemen geringer sein als zwei TB.

- Auf dem Root-Pool kann die Komprimierung aktiviert werden, allerdings erst nach Installation des Root-Pools. Während der Installation eines Root-Pools kann die Komprimierung nicht aktiviert werden. Der Komprimierungs-Algorithmus `gzip` wird auf Root-Pools nicht unterstützt.
- Benennen Sie den Root-Pool nicht um, nachdem dieser in einer Erstinstallation erstellt wurde oder nachdem eine Solaris Live Upgrade-Migration auf ein ZFS-Root-Dateisystem durchgeführt wurde. Das Umbenennen des Root-Pools kann dazu führen, dass das System nicht gebootet werden kann.

  Ändern Sie außerdem den Standardeinhängepunkt der Root-Poolkomponenten nicht, wenn Sie Live Upgrade verwenden möchten.
- Wenn Sie Ihre Swap- und Dump-Geräte ändern möchten und Live Upgrade verwenden, wird auf „Anpassen der ZFS-Swap- und Dump-Volumes“ auf Seite 161 verwiesen.

Installieren eines ZFS-Root-Dateisystems (Erstinstallation von Oracle Solaris)

In dieser Oracle Solaris-Version stehen Ihnen folgende Verfahren für die Erstinstallation zur Verfügung:

- Verwenden Sie das interaktive textbasierte Installationsprogramm zur Erstinstallation eines ZFS-Speicher-Pools mit einem bootfähigen ZFS-Root-Dateisystem. Wenn Sie einen bereits vorhandenen ZFS-Speicher-Pool für das ZFS-Root-Dateisystem verwenden möchten, müssen Sie das vorhandene UFS-Root-Dateisystem mit Live Upgrade in ein ZFS-Root-Dateisystem in einem vorhandenen ZFS-Speicher-Pool migrieren. Weitere Informationen finden Sie unter „Migrieren in ein ZFS-Root-Dateisystem oder Aktualisieren eines ZFS-Root-Dateisystems (Live Upgrade)“ auf Seite 132.
- Verwenden Sie das interaktive textbasierte Installationsprogramm zur Erstinstallation eines ZFS-Speicher-Pools mit einem bootfähigen ZFS-Root-Dateisystem aus einem ZFS-Flash-Archiv.

Bevor Sie mit der Erstinstallation für die Erstellung eines ZFS-Speicherpools beginnen, lesen Sie den Abschnitt „Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung“ auf Seite 113.

Informationen zum Konfigurieren von Zonen und zum Patchen oder Aktualisieren des Systems nach der Erstinstallation eines ZFS-Root-Dateisystems finden Sie unter „Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)“ auf Seite 142 oder „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147.

Sollten bereits ZFS-Speicher-Pools auf dem System vorhanden sein, werden sie durch die folgende Meldung bestätigt. Diese Pools bleiben jedoch unverändert, es sei denn, Sie wählen die Festplatten in den vorhandenen Pools aus, um den neuen Speicher-Pool zu erstellen.

```
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.
```

Achtung – Vorhandene Pools werden überschrieben, falls einige ihrer Datenträger für den neuen Pool ausgewählt werden.

BEISPIEL 4–1 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems

Der interaktive, textgestützte Installationsprozess ist im Wesentlichen derselbe wie in den vorherigen Oracle Solaris-Versionen, mit der Ausnahme, dass Sie gefragt werden, ob ein UFS- oder ZFS-Root-Dateisystem erstellt werden soll. UFS ist auch in dieser Version weiterhin das Standarddateisystem. Wenn Sie ein ZFS-Root-Dateisystem wählen, werden Sie aufgefordert, einen ZFS-Speicher-Pool zu erstellen. Es folgen die Schritte zur Installation eines ZFS-Root-Dateisystems:

1. Legen Sie das Installationsmedium von Oracle Solaris ein oder booten Sie das System von einem Installationsserver. Wählen Sie dann das interaktive textbasierte Installationsverfahren zur Erstellung eines bootfähigen ZFS-Root-Dateisystems.
   - SPARC: Verwenden Sie die folgende Syntax für die Oracle Solaris-Installations-DVD:

     ```
     ok boot cdrom - text
     ```
   - SPARC: Verwenden Sie die folgende Syntax beim Booten über das Netzwerk:

     ```
     ok boot net - text
     ```
   - x86: Wählen Sie das Textmodus-Installationsverfahren.

   Sie können auch ein ZFS-Flash-Archiv erstellen, das mit folgenden Methoden installiert werden kann:
   - JumpStart-Installation. Weitere Informationen finden Sie in Beispiel 4–2.
   - Erstinstallation. Weitere Informationen finden Sie in Beispiel 4–3.

   Sie können ein Standard-Upgrade ausführen, um ein vorhandenes bootfähiges ZFS-Dateisystem zu aktualisieren. Ein neues bootfähiges ZFS-Dateisystem kann mit dieser Option jedoch nicht erstellt werden. Ab der Solaris-Version 10 10/08 können Sie ein UFS-Root-Dateisystem in ein ZFS-Root-Dateisystem migrieren. Hierfür muss mindestens Solaris 10 10/08 installiert sein. Weitere Informationen zur Migration in ein ZFS-Root-Dateisystem finden Sie unter „Migrieren in ein ZFS-Root-Dateisystem oder Aktualisieren eines ZFS-Root-Dateisystems (Live Upgrade)“ auf Seite 132.
2. Wenn Sie ein ZFS-Root-Dateisystem erstellen möchten, wählen Sie die ZFS-Option. Beispiel:

   ```
   Choose Filesystem Type

     Select the filesystem to use for your Solaris installation
   ```

BEISPIEL 4–1 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems *(Fortsetzung)*

```
            [ ] UFS
            [X] ZFS
```

3. Nach der Auswahl der zu installierenden Software werden Sie zur Auswahl der Festplatten aufgefordert, auf denen der ZFS-Speicherpool erstellt werden soll. Dieser Bildschirm sieht ähnlich aus wie in vorherigen Versionen.

```
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                                            139989 MB  (F4 to edit
) [ ]    c1t1d0                                            139989 MB
  [ ]    c1t2d0                                            139989 MB
  [ ]    c1t3d0                                            139989 MB
  [ ]    c2t0d0                                            139989 MB
  [ ]    c2t1d0                                            139989 MB
  [ ]    c2t2d0                                            139989 MB
  [ ]    c2t3d0                                            139989 MB
                                  Maximum Root Size: 139989 MB
                                  Suggested Minimum:  11102 MB
```

Sie können mindestens eine Festplatte für den ZFS-Root-Pool auswählen. Wenn Sie zwei Festplatten auswählen, wird für den Root-Pool eine aus zwei Festplatten bestehende Konfiguration mit Datenspiegelung ausgewählt. Optimal ist ein gespiegelter Pool mit zwei oder drei Festplatten. Wenn Sie über acht Festplatten verfügen und alle auswählen, werden diese acht Festplatten als eine große Datenspiegelung für den Root-Pool verwendet. Diese Konfiguration ist nicht optimal. Alternativ können Sie einen gespiegelten Root-Pool nach der Erstinstallation erstellen. RAID-Z-Konfigurationen werden für den Root-Pool nicht unterstützt.

Weitere Informationen zur Konfiguration von ZFS-Speicher-Pools finden Sie unter „Replikationsfunktionen eines ZFS-Speicher-Pools“ auf Seite 49.

4. Um zwei Festplatten zur Erstellung eines gespiegelten Root-Pools auszuwählen, wählen Sie die zweite Festplatte mit dem Cursor aus.

Im folgenden Beispiel werden sowohl `c1t0d0` als auch `c1t1d0` als Root-Pool-Festplatten ausgewählt. Beide Festplatten müssen ein SMI-Label und den Bereich 0 haben. Falls die Festplatten kein SMI-Label haben oder Bereiche enthalten, verlassen Sie das Installationsprogramm und verwenden das Serviceprogramm `format` zur Umbenennung und Neupartitionierung der Festplatten. Anschließend starten Sie das Installationsprogramm erneut.

```
Select Disks
  On this screen you must select the disks for installing Solaris software.
```

BEISPIEL 4–1 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems *(Fortsetzung)*

```
  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                                              139989 MB  (F4 to edit
) [X]    c1t1d0                                              139989 MB
  [ ]    c1t2d0                                              139989 MB
  [ ]    c1t3d0                                              139989 MB
  [ ]    c2t0d0                                              139989 MB
  [ ]    c2t1d0                                              139989 MB
  [ ]    c2t2d0                                              139989 MB
  [ ]    c2t3d0                                              139989 MB

                                  Maximum Root Size: 139989 MB
                                  Suggested Minimum:  11102 MB
```

Wenn in der Spalte "Verfügbarer Speicherplatz" 0 MB angezeigt werden, bedeutet dies in der Regel, dass die Festplatte ein EFI-Label hat. Wenn Sie eine Festplatte mit einem EFI-Label verwenden möchten, verlassen Sie das Installationsprogramm, versehen die Festplatte mit einem neuen SMI-Label, indem Sie den Befehl `format -e` verwenden, und starten anschließend das Installationsprogramm erneut.

Wenn Sie während der Installation keinen gespiegelten Root-Pool erstellen, können Sie dies nach der Installation problemlos nachholen. Weitere Informationen finden Sie unter „Erstellen eines gespiegelten ZFS-Root-Pools (nach der Installation)“ auf Seite 122.

Nach der Auswahl mindestens einer Festplatte für den ZFS-Speicher-Pool wird ein Bildschirm wie der folgende angezeigt:

```
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: s10nameBE
        ZFS Pool Size (in MB): 139990
  Size of Swap Area (in MB): 4096
  Size of Dump Area (in MB): 1024
        (Pool size must be between 7006 MB and 139990 MB)

                          [X] Keep / and /var combined
                          [ ] Put /var on a separate dataset
```

5. Auf diesem Bildschirm können Sie optional den Namen des ZFS-Pools, den Dataset-Namen, die Pool-Größe sowie die Größe der Swap- und Dump-Geräte ändern. Gehen Sie mit dem Cursor von Eintrag zu Eintrag und ersetzen Sie die Standardwerte durch

BEISPIEL 4–1 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems *(Fortsetzung)*

die neuen Werte. Anderenfalls können Sie die Standardwerte übernehmen. Außerdem können Sie das Verfahren ändern, mit dem das Dateisystem `/var` erstellt und eingehängt wird.

In diesem Beispiel wird der Name des Root-Datasets in `zfsBE` abgeändert.

```
             ZFS Pool Name: rpool
     ZFS Root Dataset Name: zfsBE
     ZFS Pool Size (in MB): 139990
  Size of Swap Area (in MB): 4096
  Size of Dump Area (in MB): 1024
       (Pool size must be between 7006 MB and 139990 MB)
```

6. Auf diesem letzten Installationsbildschirm können Sie optional das Installationsprofil ändern. Beispiel:

```
Profile

  The information shown below is your profile for installing Solaris software.
  It reflects the choices you've made on previous screens.

  ============================================================================

               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: 139990 MB
                   Devices in Pool: c1t0d0
                                    c1t1d0
```

7. Überprüfen Sie nach abgeschlossener Installation die Informationen zum erstellten ZFS-Speicher-Pool und Dateisystem. Beispiel:

```
# 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
```

BEISPIEL 4–1 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems *(Fortsetzung)*

```
rpool              10.1G   124G   106K  /rpool
rpool/ROOT         5.01G   124G    31K  legacy
rpool/ROOT/zfsBE   5.01G   124G  5.01G  /
rpool/dump         1.00G   124G  1.00G  -
rpool/export         63K   124G    32K  /export
rpool/export/home    31K   124G    31K  /export/home
rpool/swap         4.13G   124G  4.00G  -
```

In der Beispielausgabe von `zfs list` sind die Root-Pool-Komponenten wie z. B. das Verzeichnis `rpool/ROOT` aufgeführt, auf das standardmäßig nicht zugegriffen werden kann.

8. Um eine weitere ZFS-Boot-Umgebung (BU) im selben Speicher-Pool zu erstellen, verwenden Sie den Befehl `lucreate`.

   Im folgenden Beispiel wird eine neue BU namens `zfs2BE` erstellt. Die aktuelle BU wird mit `zfsBE` benannt, wie in der Ausgabe von `zfs list` gezeigt wird. Die aktuelle BU wird jedoch erst nach der Erstellung der neuen BU in der Ausgabe von `lustatus` bestätigt.

   ```
   # lustatus
   ERROR: No boot environments are configured on this system
   ERROR: cannot determine list of all boot environment names
   ```

   Zum Erstellen einer neuen ZFS-BU in demselben Pool verwenden Sie folgende Syntax:

```
# 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.
```

   Beim Erstellen einer ZFS-BU innerhalb desselben Pools kommen die Klon- und Schnappschuss-Funktionen von ZFS zum Einsatz und die BU wird sofort erstellt. Weitere Informationen zur ZFS-Root-Migration mithilfe von Live Upgrade finden Sie unter „Migrieren in ein ZFS-Root-Dateisystem oder Aktualisieren eines ZFS-Root-Dateisystems (Live Upgrade)“ auf Seite 132.

9. Überprüfen Sie anschließend die neuen Boot-Umgebungen. Beispiel:

   ```
   # lustatus
   Boot Environment           Is       Active Active    Can    Copy
   ```

BEISPIEL 4–1 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems *(Fortsetzung)*

```
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                    10.1G   124G   106K  /rpool
rpool/ROOT               5.00G   124G    31K  legacy
rpool/ROOT/zfs2BE         218K   124G  5.00G  /
rpool/ROOT/zfsBE         5.00G   124G  5.00G  /
rpool/ROOT/zfsBE@zfs2BE   104K      -  5.00G  -
rpool/dump               1.00G   124G  1.00G  -
rpool/export               63K   124G    32K  /export
rpool/export/home          31K   124G    31K  /export/home
rpool/swap               4.13G   124G  4.00G  -
```

10. Zum Booten aus einer alternativen BU verwenden Sie den Befehl `luactivate`.

    - SPARC – Ermitteln Sie mit dem Befehl `boot -L` die verfügbaren BUs, wenn das Boot-Gerät einen ZFS-Speicher-Pool enthält.

      Geben Sie also beispielsweise bei einem SPARC-System den Befehl `boot -L` ein, um eine Liste der verfügbaren BUs anzuzeigen. Zum Booten aus der neuen BU `zfs2BE` wählen Sie Option 2. Geben Sie dann den angezeigten Befehl `boot -Z` ein.

```
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
```

    - x86 – Ermitteln Sie die über das GRUB-Menü zu bootende BU.

Weitere Informationen zum Booten eines ZFS-Dateisystems können Sie dem Abschnitt „Booten aus einem ZFS-Root-Dateisystem“ auf Seite 162 entnehmen.

▼ Erstellen eines gespiegelten ZFS-Root-Pools (nach der Installation)

Wenn Sie während der Installation keinen gespiegelten ZFS-Root-Pool erstellen, können Sie dies nach der Installation problemlos nachholen.

Informationen zum Ersetzen einer Festplatte in einem ZFS-Root-Pool finden Sie unter „So ersetzen Sie eine Festplatte im ZFS-Root-Pool“ auf Seite 170.

1 Zeigen Sie den aktuellen Root-Pool-Status an.

```
# 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 Binden Sie eine zweite Festplatte ein, um einen gespiegelten Root-Pool zu konfigurieren.

```
# zpool attach rpool c1t0d0s0 c1t1d0s0
Make sure to wait until resilver is done before rebooting.
```

3 Zeigen Sie den Root-Pool-Status an, um zu bestätigen, dass das Resilvering abgeschlossen ist.

```
# 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
```

In der vorhergehenden Ausgabe ist das Resilvering nicht abgeschlossen. Das Resilvering ist abgeschlossen, wenn eine Meldung wie die folgende angezeigt wird:

```
resilvered 10.0G in 0h10m with 0 errors on Thu Nov 15 12:48:33 2012
```

4 Überprüfen Sie, ob Sie von der zweiten Festplatte booten können.

5 Richten Sie bei Bedarf das System so ein, dass es automatisch von der neuen Festplatte gebootet wird.

- SPARC – Verwenden Sie den Befehl `eeprom` oder `setenv` am SPARC-Boot-PROM, um das Standard-Boot-Gerät zurückzusetzen.
- x86 – Konfigurieren Sie das System-BIOS erneut.

Installieren eines ZFS-Root-Dateisystems (Oracle Solaris Flash-Archiv-Installation)

In Solaris 10 10/09 kann ein Flash-Archiv auf einem System mit einem UFS-Root-Dateisystem oder einem ZFS-Root-Dateisystem erstellt werden. Ein Flash-Archiv eines ZFS-Root-Pools enthält die gesamte Pool-Hierarchie außer Swap- und Dump-Volumes und ausgeschlossene Datensätze. Die Swap- und Dump-Volumes werden bei der Installation des Flash-Archivs erstellt. Sie können bei der Flash-Archiv-Installation wie folgt vorgehen:

- Erstellen Sie ein Flash-Archiv, das zur Installation und zum Starten eines Systems mit einem ZFS-Root-Dateisystem verwendet werden kann.
- Führen Sie eine JumpStart- oder Erstinstallation eines Klonsystems unter Verwendung eines ZFS-Flash-Archivs aus. Durch die Erstellung eines ZFS-Flash-Archivs wird ein ganzer Root-Pool geklont, nicht nur einzelne Boot-Umgebungen. Einzelne Datasets innerhalb des Pools können mit der Option `-D` der Befehle `flarcreate` und `flar` ausgeschlossen werden.

Beachten Sie folgende Einschränkungen, bevor Sie ein System mit einem ZFS-Flash-Archiv installieren:

- Ab Oracle Solaris 10 8/11 können Sie mithilfe der Flash-Archiv-Option der interaktiven Installation ein System mit einem ZFS-Root-Dateisystem installieren. Darüber hinaus können Sie mit einem Flash-Archiv eine alternative ZFS-BU aktualisieren, indem Sie den Befehl `luupgrade` verwenden.
  - Bei einem System, auf dem Solaris 10 9/10 ausgeführt wird, muss Patch 124630-51 (SPARC) oder Patch 124631-51 (x86) hinzugefügt werden, um ein Flash-Archiv in einem ABE zu installieren.
  - Das Master-System, in dem Sie das Flash-Archiv erstellen, und das Klonsystem, in dem Sie das Flash-Archiv installieren, müssen sich auf derselben Kernel-Patchebene befinden. Beispiel: Wenn Sie ein ZFS-Flash-Archiv auf einem System erstellen, auf dem Solaris 10 8/11 läuft, muss das Klonsystem ebenfalls auf derselben Solaris 10 8/11-Kernel-Patchebene ausgeführt werden. Andernfalls kann die Installation des Flash-Archivs mit Fehlern bei dem Befehl `zfs receive` nicht erfolgreich verlaufen.
  - Ein Master-System, das unter Solaris 10 9/10 mit untergeordneten Root-Dateisystemen läuft, wie einem separaten `/var`-Dateisystem, muss auf die Solaris 10 8/11-Version upgegradet werden, bevor das Flash-Archiv erstellt und für ABE angewendet wird. Sonst verläuft die Installation des Flash-Archivs nicht erfolgreich.
- Sie können ein Flash-Archiv nur auf einem System installieren, das dieselbe Architektur wie das System besitzt, auf dem Sie das ZFS-Flash-Archiv erstellt haben. Beispielsweise kann ein auf einem `sun4v`-System erstelltes Archiv nicht auf einem `sun4u`-System installiert werden.
- Es wird nur die vollständige Erstinstallation eines ZFS-Flash-Archivs unterstützt. Sie können weder ein anderes Flash-Archiv eines ZFS-Root-Dateisystems noch ein Hybrid-UFS/ZFS-Archiv installieren.

- Ab Solaris 10 8/11 können Sie mit einem UFS-Flash-Archiv ein ZFS-Root-Dateisystem installieren. Beispiel:
  - Wenn Sie das Schlüsselwort `pool` im JumpStart-Profil verwenden, wird das UFS-Flash-Archiv in einem ZFS-Root-Pool installiert.

    ```
    pool rpool auto auto auto mirror c0t0d0s0 c0t1d0s0
    ```
  - Wählen Sie bei der interaktiven Installation eines UFS-Flash-Archivs ZFS als Dateisystemtyp aus.
- Auch wenn der gesamte Root-Pool, ausdrücklich ausgeschlossene Datasets ausgenommen, archiviert und installiert wird, ist nur die ZFS-BU, die bei Erstellung des Archivs gebootet wird, nach Installation des Flash-Archivs verwendbar. Pools, die mit der Option `-R` *rootdir* des Befehls `flarcreate` oder `flar` archiviert wurden, können jedoch zur Archivierung eines anderen als des aktuell gebooteten Root-Pools verwendet werden.
- Die Befehlsoptionen `flarcreate` und `flar` zum Ein- und Ausschließen einzelner Dateien werden in einem ZFS-Flash-Archiv nicht unterstützt. Sie können nur komplette Datasets aus einem ZFS-Flash-Archiv ausschließen.
- Der Befehl `flar info` wird für ein ZFS-Flash-Archiv nicht unterstützt. Beispiel:

  ```
  # flar info -l zfs10upflar
  ERROR: archive content listing not supported for zfs archives.
  ```

Nachdem ein Mastersystem mit Solaris 10 10/09 oder einer höheren Version installiert oder aktualisiert wurde, können Sie ein ZFS-Flash-Archiv erstellen, um ein Zielsystem zu installieren. Dieser Vorgang läuft im Wesentlichen wie folgt ab:

- Erstellen Sie das ZFS-Flash-Archiv mit dem Befehl `flarcreate` auf dem Mastersystem. Alle Datasets im Root-Pool außer Swap- und Dump-Volumes werden im ZFS-Flash-Archiv beinhaltet.
- Erstellen Sie ein JumpStart-Profil, um die Flash-Archiv-Information auf dem Installationsserver zu beinhalten.
- Installieren Sie das ZFS-Flash-Archiv auf dem Zielsystem.

Die folgenden Archivierungsoptionen werden zur Installation eines ZFS-Root-Pools mit einem Flash-Archiv unterstützt:

- Verwenden Sie den Befehl `flarcreate` oder `flar`, um ein Flash-Archiv aus dem angegebenen ZFS-Root-Pool zu erstellen. Falls kein ZFS-Root-Pool angegeben ist, wird ein Flash-Archiv aus dem Standard-Root-Pool erstellt.
- Verwenden Sie `flarcreate -D` *dataset*, um das angegebene Dataset aus dem Flash-Archiv auszuschließen. Diese Option kann mehrmals zum Ausschließen mehrerer Datasets verwendet werden.

Nach Installation eines ZFS-Flash-Archivs wird das System wie folgt konfiguriert:

- Die komplette auf dem System, auf dem das Flash-Archiv erstellt wurde, vorhandene Dataset-Hierarchie wird auf dem Zielsystem neu erstellt, ausgenommen jegliche Datasets, die bei der Erstellung des Archivs ausgeschlossen wurden. Die Swap- und Dump-Volumes werden nicht in das Flash-Archiv eingeschlossen.
- Der Root-Pool hat den gleichen Namen wie der Pool, aus dem das Archiv erstellt wurde.
- Die zum Zeitpunkt der Erstellung des Flash-Archivs aktive BU ist die aktive und standardmäßige BU auf den verwendeten Systemen.

BEISPIEL 4–2 Installieren eines ZFS-Flash-Archivs auf einem System (JumpStart-Installation)

Nachdem Sie das Mastersystem mit Solaris 10 10/09 oder einer höheren Version installiert oder aktualisiert haben, erstellen Sie ein Flash-Archiv des ZFS-Root-Pools. Beispiel:

```
# flarcreate -n zfsBE zfs10upflar
Full Flash
Checking integrity...
Integrity OK.
Running precreation scripts...
Precreation scripts done.
Determining the size of the archive...
The archive will be approximately 6.77GB.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.

Running pre-exit scripts...
Pre-exit scripts done.
```

Erstellen Sie anschließend auf dem System, das als Installationsserver dienen soll, ein JumpStart-Profil. Gehen Sie dabei so wie bei der normalen Installation eines Systems vor. Das folgende Profil wird beispielsweise für die Installation des `zfs10upflar`-Archivs verwendet:

```
install_type flash_install
archive_location nfs system:/export/jump/zfs10upflar
partitioning explicit
pool rpool auto auto auto mirror c0t1d0s0 c0t0d0s0
```

BEISPIEL 4–3 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems (Flash-Archiv-Installation)

Sie können ein ZFS-Root-Dateisystem durch Auswahl der Option zur Flash-Installation installieren. Bei dieser Option wird davon ausgegangen, dass bereits ein ZFS-Flash-Archiv erstellt wurde und zur Verfügung steht.

1. Wählen Sie auf dem Bildschirm des interaktiven Solaris-Installationsprogramms die F4_Flash-Option.
2. Wählen Sie auf dem Bildschirm "Neustart nach der Installation?" die Option "Automatischer Neustart" oder "Manueller Neustart".
3. Wählen Sie auf dem Bildschirm "Auswählen von Dateisystemtyp" die Option "ZFS".

BEISPIEL 4–3 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems (Flash-Archiv-Installation) *(Fortsetzung)*

4. Wählen Sie auf dem Bildschirm "Abrufmethode für Flash-Archive" die Abrufmethode, wie z. B. "HTTP", "FTP", "NFS", "Lokale Datei", "Lokales Band" oder "Lokales Gerät".

   Wählen Sie z. B. "NFS", wenn das ZFS-Flash-Archiv über einen NFS-Server gemeinsam genutzt wird.

5. Geben Sie auf dem Bildschirm "Hinzufügen von Flash-Archiven" den Speicherort des ZFS-Flash-Archivs an.

   Handelt es sich beim Speicherort beispielsweise um einen NFS-Server, ermitteln Sie den Server anhand von dessen IP-Adresse und geben Sie dann den Pfad zum ZFS-Flash-Archiv an.

   ```
   NFS Location: 12.34.567.890:/export/zfs10upflar
   ```

6. Bestätigen Sie auf dem Bildschirm "Auswahl von Flash-Archiven" die Abrufmethode und den Namen der ZFS-BU.

   ```
   Flash Archive Selection

     You selected the following Flash archives to use to install this system.  If
     you want to add another archive to install select "New".

           Retrieval Method      Name
           ========================================================================
           NFS                   zfsBE
   ```

7. Überprüfen Sie die nachfolgenden Bildschirme wie bei einer Erstinstallation und wählen Sie die Ihrer Konfiguration entsprechenden Optionen:

   - Platten auswählen
   - Daten beibehalten?
   - Konfigurieren der ZFS-Einstellungen

     Überprüfen Sie die Zusammenfassungsinformationen und wählen Sie dann die Option "Weiter".

     Beispiel:

     ```
     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: s10zfsBE
           ZFS Pool Size (in MB): 69995
       Size of Swap Area (in MB): 2048
       Size of Dump Area (in MB): 1024
             (Pool size must be between 7591 MB and 69995 MB)
     ```

BEISPIEL 4–3 Erstinstallation eines bootfähigen ZFS-Root-Dateisystems (Flash-Archiv-Installation) *(Fortsetzung)*

Handelt es sich beim Flash-Archiv um einen ZFS send-Datenstrom, werden die kombinierten oder separaten `/var`-Dateisystemoptionen nicht dargestellt. In diesem Fall hängt es von der Konfiguration von `/var` auf dem Mastersystem ab, ob das Dateisystem kombiniert wird oder nicht.

- Klicken Sie auf dem Bildschirm "Entfernte Dateisysteme einhängen?" auf "Weiter".
- Überprüfen Sie den Profilbildschirm. Drücken Sie F4, um Änderungen vorzunehmen. Verwenden Sie andernfalls die Option "Begin_Installation" (F2).

  Beispiel:

```
Profile

  The information shown below is your profile for installing Solaris software.
  It reflects the choices you've made on previous screens.

  ============================================================================

              Installation Option: Flash
                        Boot Device: c1t0d0
                Root File System Type: ZFS
                      Client Services: None

                             Software: 1 Flash Archive
                                       NFS: zfsBE
                           Pool Name: rpool
                Boot Environment Name: s10zfsBE
                           Pool Size: 69995 MB
                      Devices in Pool: c1t0d0
```

Installieren eines ZFS-Root-Dateisystems (JumpStart-Installation)

Sie können eine JumpStart-Profil erstellen, um ein ZFS- oder UFS-Root-Dateisystem zu installieren.

ZFS-spezifische JumpStart-Profile müssen das neue Schlüsselwort `pool` enthalten. Mit dem Schlüsselwort `pool` wird ein neuer Root-Pool installiert. Dabei wird standardmäßig eine neue Boot-Umgebung (BU) erstellt. Sie haben die Möglichkeit, einen Namen für die BU anzugeben und ein separates `/var`-Dataset zu erstellen. Hierzu dienen die Schlüsselwörter `bootenv installbe` und die Optionen `bename` und `dataset`.

Allgemeine Informationen zu den JumpStart-Funktionen finden Sie in *Oracle Solaris 10 1/13 Installationshandbuch: JumpStart-Installation*.

Informationen zum Konfigurieren von Zonen und zum Patchen oder Aktualisieren des Systems nach der JumpStart-Installation eines ZFS-Root-Dateisystems finden Sie unter

„Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)“ auf Seite 142 oder „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147.

JumpStart-Schlüsselwörter für ZFS

Die folgenden Schlüsselwörter sind in ZFS-spezifischen JumpStart-Profilen zulässig:

`auto` Legt automatisch die Größe der Bereiche für Pool, Swap-Volume oder Dump-Volume fest. Die Größe der Festplatte wird auf Einhaltung der Mindestgröße überprüft. Wenn die Mindestgröße verfügbar ist, wird die im Rahmen der gegebenen Einschränkungen (Größe der Festplatten, reservierte Bereiche usw.) maximale Pool-Größe zugewiesen.

Wenn Sie beispielsweise `c0t0d0s0` und eines der Schlüsselwörter `all` oder `auto` angeben, wird der Root-Pool-Bereich so groß wie möglich erstellt. Sie können aber auch eine bestimmte Größe für den Bereich, das Swap-Volume oder das Dump-Volume angeben.

Im Zusammenhang mit einem ZFS-Root-Pool verhält sich das Schlüsselwort `auto` wie das Schlüsselwort `all`, da Pools keine ungenutzte Festplattenkapazität haben.

`bootenv` Mit diesem Schlüsselwort werden die Merkmale der Boot-Umgebung festgelegt.

Zum Erstellen einer boot-fähigen ZFS-Root-Umgebung verwenden Sie das Schlüsselwort `bootenv` mit der folgenden Syntax:

`bootenv installbe bename` *BU-Name* [*dataset Einhängepunkt*]

`installbe` Erstellt und installiert eine neue BU, die durch die Option `bename` und den Eintrag *BE-name* bestimmt wird.

`bename` *BU-Name* Gibt den *BU-Namen* für die Installation an.

Wenn Sie `bename` nicht zusammen mit dem Schlüsselwort `pool` angeben, wird eine Standard-BU erstellt.

`dataset` *mount-point* Das optionale Schlüsselwort `dataset` ermöglicht es, ein vom Root-Dataset separates `/var`-Dataset anzugeben. Der Wert für *Einhängepunkt* ist derzeit auf `/var` beschränkt. So würde beispielsweise eine `bootenv`-Syntaxzeile für ein separates `/var`-Dataset wie folgt aussehen:

```
bootenv installbe bename zfsroot dataset /var
```

`pool` Definiert den neuen, zu erstellenden Root-Pool. Die folgende Schlüsselwortsyntax muss verwendet werden:

`pool` *poolname poolsize swapsize dumpsize vdevlist*

Pool-Name Gibt den Namen des zu erstellenden Pools an. Der Pool wird mit der angegebenen Größe *poolsize* und den physischen Geräten erstellt, die für ein oder mehrere Geräte angegeben sind (*vdevlist*). Geben Sie für den Wert *poolname* nicht den Namen eines vorhandenen Pools an, da sonst der vorhandene Pool überschrieben wird.

Pool-Größe Gibt die Größe des zu erstellenden Pools an. Der Wert kann `auto` oder `existing` sein. Mit dem Wert `auto` wird die im Rahmen der gegebenen Einschränkungen (z. B. Größe der Festplatten usw.) maximale Pool-Größe zugewiesen. Sofern nicht `g` (GB) angegeben wird, gilt für die Größenangabe die Einheit MB.

Swap-Größe Gibt die Größe des zu erstellenden Swap-Volumes an. Der Wert `auto` bewirkt, dass die Standard-Swap-Größe verwendet wird. Sie können eine Größe angeben, indem Sie einen *size*-Wert verwenden. Sofern nicht `g` (GB) angegeben wird, gilt für die Größenangabe die Einheit MB.

Dump-Größe Gibt die Größe des zu erstellenden Dump-Volumes an. Der Wert `auto` bewirkt, dass die Standard-Dump-Größe verwendet wird. Sie können eine Größe angeben, indem Sie einen *size*-Wert verwenden. Sofern nicht `g` (GB) angegeben wird, gilt für die Größenangabe die Einheit MB.

vdev-Liste Gibt eines oder mehrere Speichergeräte an, aus denen der Pool gebildet wird. Das Format der *vdev-Liste* ist identisch mit dem Format für den Befehl `zpool create`. Derzeit werden bei der Angabe mehrerer Geräte nur gespiegelte Konfigurationen unterstützt. Bei den Geräten in *vdevlist* muss es sich um Bereiche für den Root-Pool handeln. Der Wert `any` bewirkt, dass die Installationssoftware ein passendes Gerät wählt.

Sie können beliebig viele Festplatten spiegeln. Die Größe des erstellten Pools richtet sich jedoch nach der kleinsten der angegebenen Festplatten. Weitere Informationen zum Erstellen von gespiegelten Speicher-Pools finden Sie unter „Speicher-Pools mit Datenspiegelung“ auf Seite 49.

JumpStart-Profilbeispiele für ZFS

Dieser Abschnitt enthält Beispiele für ZFS-spezifische JumpStart-Profile.

Das folgende Profil führt eine Erstinstallation (angegeben mit `install_type initial_install`) in einem neuen Pool (angegeben mit `pool newpool`) durch, dessen Größe gemäß dem Schlüsselwort `auto` automatisch auf die Größe der angegebenen Festplatten bemessen wird. Die Größe des Swap-Bereichs und des Dump-Geräts in einer Festplattenkonfiguration mit Datenspiegelung wird automatisch mit dem Schlüsselwort `auto` festgelegt (mit dem Schlüsselwort `mirror` und Angabe der Datenträger als `c0t0d0s0` und `c0t1d0s0`). Die Merkmale der Boot-Umgebung werden mit dem Schlüsselwort `bootenv` festgelegt, eine neue BU mit dem Schlüsselwort `installbe` installiert und eine BU mit dem Wert `s10-xx` wird erstellt.

```
install_type initial_install
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename s10-xx
```

Das folgende Profil führt eine Erstinstallation mit dem Schlüsselwort `install_type initial_install` des Metaclusters `SUNWCall` in einem neuen Pool namens `newpool` durch, dessen Größe 80 GB beträgt. Dieser Pool wird mit einem Swap-Volume von 2 GB und einem Dump-Volume von ebenfalls 2 GB in einer gespiegelten Konfiguration zwei beliebiger verfügbarer Geräte erstellt, die groß genug sind, um ein Pool mit einer Kapazität von 80 GB zu erstellen. Sollten zwei derartige Geräte nicht verfügbar sein, schlägt die Installation fehl. Die Merkmale der Boot-Umgebung werden mit dem Schlüsselwort `bootenv` festgelegt, eine neue BU mit dem Schlüsselwort `installbe` installiert und der `bename` mit dem Wert *s10-xx* wird erstellt.

```
install_type initial_install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename s10-xx
```

Die JumpStart-Installationssyntax ermöglicht Ihnen, ein UFS-Dateisystem auf einem Datenträger, der auch einen ZFS-Root-Pool enthält, zu erhalten oder zu erstellen. Diese Konfiguration wird nicht für Produktionssysteme empfohlen. Sie kann jedoch für die Erfordernisse der Umstellung kleinerer Systeme (beispielsweise eines Laptops) verwendet werden.

JumpStart-Probleme im Zusammenhang mit ZFS

Vor Beginn einer JumpStart-Installation eines bootfähigen ZFS-Root-Dateisystems sind folgende Probleme zu berücksichtigen.

- Ein vorhandener ZFS-Speicher-Pool kann nicht für eine JumpStart-Installation zum Erstellen eines bootfähigen ZFS-Root-Dateisystems verwendet werden. Sie müssen einen neuen ZFS-Speicherpool mit einer Syntax wie der folgenden erstellen:

  ```
  pool rpool 20G 4G 4G c0t0d0s0
  ```

- Sie müssen den Pool mit Festplattenbereichen anstatt mit gesamten Festplatten erstellen. Siehe hierzu „Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung“ auf Seite 113. Die im folgenden Beispiel gezeigte fett gedruckte Syntax ist unzulässig:

  ```
  install_type initial_install
  cluster SUNWCall
  pool rpool all auto auto mirror c0t0d0 c0t1d0
  bootenv installbe bename newBE
  ```

  Die im folgenden Beispiel gezeigte fett gedruckte Syntax ist zulässig:

  ```
  install_type initial_install
  cluster SUNWCall
  pool rpool all auto auto mirror c0t0d0s0 c0t1d0s0
  bootenv installbe bename newBE
  ```

Migrieren in ein ZFS-Root-Dateisystem oder Aktualisieren eines ZFS-Root-Dateisystems (Live Upgrade)

Die Live Upgrade-Funktionen aus früheren Versionen sind weiterhin verfügbar und verhalten so wie in den vorherigen Versionen auch.

Folgende Leistungsmerkmale sind verfügbar:

- **Migration von UFS-BU in ZFS-BU**
  - Für die Migration eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem muss mit der Option `-p` ein vorhandener ZFS-Speicher-Pool bestimmt werden.
  - Wenn das UFS-Root-Dateisystem Komponenten auf verschiedenen Bereichen umfasst, werden diese in den ZFS-Root-Pool überführt.
  - In Oracle Solaris 10 8/11 können Sie beim Migrieren des UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem ein separates `/var`-Dateisystem angeben.
  - Die Migration eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem läuft in folgenden Grundschritten ab:
    1. Installieren Sie bei Bedarf die erforderlichen Live Upgrade-Patches.
    2. Installieren Sie eine aktuelle Oracle Solaris 10-Version (Solaris 10 10/08 oder Solaris 10 8/11) oder verwenden Sie das Standard-Upgrade-Programm, um eine Aktualisierung einer früheren Solaris 10-Version auf einem beliebigen unterstützten SPARC- oder x86-System durchzuführen.

3. Wenn Sie Solaris 10 10/08 oder eine höhere Version ausführen, erstellen Sie einen ZFS-Speicher-Pool für das ZFS-Root-Dateisystem.
4. Verwenden Sie Live Upgrade, um das UFS-Root-Dateisystem in ein ZFS-Root-Dateisystem zu migrieren.
5. Aktivieren der ZFS-BU mit dem Befehl `luactivate`.

- **Patchen oder Aktualisieren einer ZFS-BU**
  - Mit dem Befehl `luupgrade` können Sie eine vorhandene ZFS-BU patchen oder aktualisieren. Ebenfalls können Sie mithilfe von `luupgrade` eine alternative ZFS-BU mit einem ZFS-Flash-Archiv aktualisieren. Weitere Informationen finden Sie in Beispiel 4–8.
  - Live Upgrade kann die ZFS-Schnappschuss- und -Klon-Funktionen verwenden, wenn Sie eine neue ZFS-BU in demselben Pool erstellen. Dadurch wird die BU-Erstellung wesentlich schneller als in vorherigen Versionen.
- **Unterstützung für Zonenmigration** – Sie können ein System mit Zonen zwar migrieren, die unterstützten Konfigurationen sind in Solaris 10 10/08 jedoch begrenzt. Ab Solaris 10 5/09 werden mehr Zonenkonfigurationen unterstützt. Weitere Informationen finden Sie in den folgenden Abschnitten:
  - „Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)“ auf Seite 142
  - „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147

  Informationen zum Migrieren eines Systems ohne Zonen finden Sie unter „Migrieren oder Aktualisieren eines ZFS-Root-Dateisystems mit Live Upgrade (ohne Zonen)“ auf Seite 135.

Ausführliche Informationen zu den Oracle Solaris-Installations- und Live Upgrade-Funktionen finden Sie im *Oracle Solaris 10 1/13 Installationshandbuch: Live Upgrade und Planung von Upgrades*.

Informationen zu Voraussetzungen für ZFS und Live Upgrade finden Sie unter „Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung“ auf Seite 113.

Probleme bei der ZFS-Migration mit Live Upgrade

Für die Migration eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem mithilfe von Live Upgrade sind folgende Probleme zu berücksichtigen:

- Die standardmäßige Upgrade-Option der grafischen Oracle Solaris-Installationsoberfläche ist für die Migration aus einem UFS-Root-Dateisystem in ein ZFS-Root-Dateisystem nicht verfügbar. Hierzu müssen Sie Live Upgrade verwenden.
- Vor dem Live Upgrade müssen Sie den ZFS-Speicher-Pool erstellen, der zum Booten verwendet werden soll. Aufgrund von derzeitigen Boot-Einschränkungen muss der ZFS-Root-Pool außerdem mit Bereichen anstatt mit ganzen Festplatten erstellt werden. Beispiel:

  ```
  # zpool create rpool mirror c1t0d0s0 c1t1d0s0
  ```

  Vergewissern Sie sich vor dem Erstellen des neuen Pools, dass die für den Pool vorgesehenen Festplatten ein SMI-Label (VTOC) anstatt eines EFI-Labels aufweisen. Bei einer Neubezeichnung der Festplatte mit einem SMI-Label ist darauf zu achten, dass durch die Bezeichnung keine Änderungen am Partitionierungsschema erfolgt sind. In den meisten Fällen sollte die gesamte Festplattenkapazität in den für den Root-Pool vorgesehenen Bereichen liegen.
- Die Erstellung einer UFS-BU aus einer ZFS-BU ist mit Oracle Solaris Live Upgrade nicht möglich. Wenn Sie eine UFS-BU in eine ZFS-BU migrieren und die UFS-BU beibehalten, kann sowohl aus der UFS-BU als auch der ZFS-BU gebootet werden.
- Benennen Sie die ZFS-BUs nicht mit dem Befehl `zfs rename` um, da Live Upgrade diese Namensänderung nicht erkennen kann. Nachfolgende Befehle wie z. B. `ludelete` schlagen fehl. Sehen Sie davon ab, ZFS-Pools oder -Dateisysteme umzubenennen, wenn Sie bereits vorhandene BUs weiterhin verwenden möchten.
- Wenn Sie eine alternative BU erstellen, die ein Klon der primären BU ist, können Sie nicht auf die Optionen `-f`, `-x`, `-y`, `-Y` und `-z` zum Einbinden oder Ausschließen von Dateien in die bzw. aus der primären BU zurückgreifen. Diese Optionen zum Einbinden und Ausschließen können weiterhin in folgenden Fällen verwendet werden:

  ```
  UFS -> UFS
  UFS -> ZFS
  ZFS -> ZFS (different pool)
  ```
- Während die Aktualisierung eines UFS-Root-Dateisystems auf ein ZFS-Root-Dateisystem mit Live Upgrade möglich ist, können Sie mit Live Upgrade keine Aktualisierung von Nicht-Root- oder freigegebenen Dateisystemen vornehmen.
- Der Befehl `lu` kann nicht zum Erstellen bzw. Migrieren eines ZFS-Root-Dateisystems verwendet werden.
- Wenn die Swap- und Dump-Geräte des Systems in einem Nicht-Root-Pool enthalten sein sollen, wird auf „Anpassen der ZFS-Swap- und Dump-Volumes“ auf Seite 161 verwiesen.

Migrieren oder Aktualisieren eines ZFS-Root-Dateisystems mit Live Upgrade (ohne Zonen)

In den nachfolgenden Beispielen wird dargestellt, wie ein UFS-Root-Dateisystem in ein ZFS-Root-Dateisystem migriert wird und wie ein ZFS-Root-Dateisystem aktualisiert wird.

Wenn Sie ein System mit Zonen migrieren oder aktualisieren, lesen Sie folgende Abschnitte:

- „Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)“ auf Seite 142
- „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147

BEISPIEL 4–4 Migrieren eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem mithilfe von Live Upgrade

Das folgende Beispiel zeigt, wie ein ZFS-Root-Dateisystem aus einem UFS-Root-Dateisystem migriert wird. Die aktuelle BU `ufsBE`, die ein UFS-Root-Dateisystem enthält, wird mit der Option `-c` angegeben. Wenn Sie die Option `-c` nicht angeben, wird als BU-Name standardmäßig der Gerätename verwendet. Die neue BU `zfsBE` wird mit der Option `-n` angegeben. Vor der Ausführung des `lucreate`-Vorgangs muss bereits ein ZFS-Speicher-Pool vorhanden sein.

Damit ein ZFS-Speicher-Pool upgrade- und bootfähig ist, muss er auf Datenträger-Speicherbereichen und darf nicht auf einer gesamten Festplatte erstellt werden. Vergewissern Sie sich vor dem Erstellen des neuen Pools, dass die für den Pool vorgesehenen Festplatten ein SMI-Label (VTOC) anstatt eines EFI-Labels aufweisen. Bei einer Neubezeichnung der Festplatte mit einem SMI-Label ist darauf zu achten, dass durch die Bezeichnung keine Änderungen am Partitionierungsschema erfolgt sind. In den meisten Fällen sollte die gesamte Festplattenkapazität in den für den Root-Pool vorgesehenen Bereichen liegen.

```
# 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>.
```

BEISPIEL 4–4 Migrieren eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem mithilfe von Live Upgrade *(Fortsetzung)*

```
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.
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.
```

Prüfen Sie nach Abschluss der `lucreate`-Vorgangs den BU-Status mithilfe des Befehls `lustatus`. Beispiel:

```
# 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    -
```

Kontrollieren Sie dann die Liste der ZFS-Komponenten. Beispiel:

```
# 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  -
```

Aktivieren Sie dann mit dem Befehl `luactivate` die neue ZFS-BU. Beispiel:

```
# 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.
```

BEISPIEL 4–4 Migrieren eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem mithilfe von Live Upgrade *(Fortsetzung)*

```
**********************************************************************
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
```

Als Nächstes starten Sie das System in der ZFS-BU neu.

```
# init 6
```

Vergewissern Sie sich, dass die ZFS-BU aktiv ist:

```
# 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     -
```

Wenn Sie wieder auf die UFS-BU zurückschalten, müssen Sie während des Bootens der ZFS-BU erstellte ZFS-Speicher-Pools zurückimportieren, da sie nicht automatisch in der UFS-BU verfügbar sind.

Wird die UFS-BU nicht mehr benötigt, kann Sie mit dem Befehl `ludelete` gelöscht werden.

BEISPIEL 4–5 Verwenden von Live Upgrade zur Erstellung einer ZFS-BU aus einer UFS-BU (mit separatem `/var`-Dateisystem)

In Oracle Solaris 10 8/11 können Sie mithilfe der Option `lucreate -D` festlegen, dass ein separates `/var`-Dateisystem erstellt werden soll, wenn Sie ein UFS-Root-Dateisystem in ein ZFS-Root-Dateisystem migrieren. Im folgenden Beispiel wird die vorhandene UFS-BU in eine ZFS-BU mit einem separaten `/var`-Dateisystem migriert.

```
# lucreate -n zfsBE -p rpool -D /var
Determining types of file systems supported
Validating file system requests
Preparing logical storage devices
Preparing physical storage devices
Configuring physical storage devices
Configuring logical storage devices
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <c0t0d0s0>.
Current boot environment is named <c0t0d0s0>.
Creating initial configuration for primary boot environment <c0t0d0s0>.
INFORMATION: No BEs are configured on this system.
The device </dev/dsk/c0t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <c0t0d0s0> PBE Boot Device </dev/dsk/c0t0d0s0>.
```

BEISPIEL 4–5 Verwenden von Live Upgrade zur Erstellung einer ZFS-BU aus einer UFS-BU (mit separatem /var-Dateisystem) *(Fortsetzung)*

```
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c0t1d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <c0t0d0s0>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Creating <zfs> file system for </var> in zone <global> on <rpool/ROOT/zfsBE/var>.
Populating file systems on boot environment <zfsBE>.
Analyzing zones.
Mounting ABE <zfsBE>.
Generating file list.
Copying data from PBE <c0t0d0s0> to ABE <zfsBE>
100% of filenames transferred
Finalizing ABE.
Fixing zonepaths in ABE.
Unmounting ABE <zfsBE>.
Fixing properties on ZFS datasets in ABE.
Reverting state of zones in PBE <c0t0d0s0>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-iaf.mnt
updating /.alt.tmp.b-iaf.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
# luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
# init 6
```

Überprüfen Sie die neu erstellten ZFS-Dateisysteme. Beispiel:

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
rpool                 6.29G  26.9G  32.5K  /rpool
rpool/ROOT            4.76G  26.9G    31K  legacy
rpool/ROOT/zfsBE      4.76G  26.9G  4.67G  /
rpool/ROOT/zfsBE/var  89.5M  26.9G  89.5M  /var
rpool/dump             512M  26.9G   512M  -
rpool/swap            1.03G  28.0G    16K  -
```

BEISPIEL 4–6 Erstellen einer ZFS-BU aus einer ZFS-BU mit Live Upgrade

Das Erstellen einer ZFS-BU aus einer ZFS-BU in demselben Pool ist ein sehr schneller Vorgang, da die ZFS-Schnappschuss- und -Klon-Funktionen dazu verwendet werden. Wenn die aktuelle BU in demselben ZFS-Pool enthalten ist, wird die Option `-p` nicht berücksichtigt.

Wenn mehrere ZFS-BUs vorhanden sind, gehen Sie wie folgt vor, um auszuwählen, aus welcher BU gebootet werden soll:

BEISPIEL 4–6 Erstellen einer ZFS-BU aus einer ZFS-BU mit Live Upgrade *(Fortsetzung)*

- SPARC: Mit dem Befehl `boot -L` können Sie die verfügbaren BUs ermitteln. Wählen Sie dann mithilfe des Befehls `boot -Z` eine BU, aus der gebootet werden soll.
- x86: Sie können eine BU aus dem GRUB-Menü auswählen.

Weitere Informationen finden Sie in Beispiel 4–12.

```
# 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.
```

BEISPIEL 4–7 Aktualisieren der ZFS-BU (`luupgrade`)

Sie können die ZFS-BU mithilfe zusätzlicher Packages oder Patches aktualisieren.

Dieser Vorgang läuft im Wesentlichen wie folgt ab:

- Erstellen Sie mit dem Befehl `lucreate` eine alternative BU.
- Aktivieren Sie die alternative BU und booten Sie.
- Fügen Sie der primären ZFS-BU mit dem Befehl `luupgrade` die Packages oder Patches hinzu.

```
# 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>.
```

BEISPIEL 4–7 Aktualisieren der ZFS-BU (`luupgrade`) *(Fortsetzung)*

```
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.
## 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.
```

Alternativ können Sie auch eine neue BU erstellen, um auf eine spätere Oracle Solaris-Version zu aktualisieren. Beispiel:

```
# luupgrade -u -n newBE -s /net/install/export/s10up/latest
```

Die Option `-s` gibt an, wo sich der Solaris-Installationsdatenträger befindet.

BEISPIEL 4–8 Erstellen einer ZFS-BU mit einem ZFS-Flash-Archiv (`luupgrade`)

In Oracle Solaris 10 8/11 können Sie mit dem Befehl `luupgrade` eine ZFS-BU aus einem vorhandenen ZFS-Flash-Archiv erstellen. Grundsätzlich sieht die Vorgehensweise folgendermaßen aus:

1. Erstellen Sie ein Flash-Archiv eines Mastersystems mit einer ZFS-BU.

   Beispiel:

   ```
   master-system# flarcreate -n s10zfsBE /tank/data/s10zfsflar
   Full Flash
   Checking integrity...
   Integrity OK.
   ```

BEISPIEL 4–8 Erstellen einer ZFS-BU mit einem ZFS-Flash-Archiv (`luupgrade`) *(Fortsetzung)*

```
Running precreation scripts...
Precreation scripts done.
Determining the size of the archive...
The archive will be approximately 4.67GB.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.

Running pre-exit scripts...
Pre-exit scripts done.
```

2. Stellen Sie das auf dem Mastersystem erstellte ZFS-Flash-Archiv für das Klonsystem zur Verfügung.

   Mögliche Speicherorte für Flash-Archive sind lokale Dateisysteme, HTTP, FTP, NFS usw.

3. Erstellen Sie eine leere alternative ZFS-BU auf dem Klonsystem.

   Geben Sie mithilfe der Option `-s -` an, dass es sich um eine leere BU handelt, die mit dem Inhalt des ZFS-Flash-Archivs aufgefüllt werden soll.

   Beispiel:

```
clone-system# lucreate -n zfsflashBE -s - -p rpool
Determining types of file systems supported
Validating file system requests
Preparing logical storage devices
Preparing physical storage devices
Configuring physical storage devices
Configuring logical storage devices
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <s10zfsBE>.
Current boot environment is named <s10zfsBE>.
Creating initial configuration for primary boot environment <s10zfsBE>.
INFORMATION: No BEs are configured on this system.
The device </dev/dsk/c0t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <s10zfsBE> PBE Boot Device </dev/dsk/c0t0d0s0>.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c0t1d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsflashBE>.
Creation of boot environment <zfsflashBE> successful.
```

4. Installieren Sie das ZFS-Flash-Archiv in der alternativen BU.

   Beispiel:

```
clone-system# luupgrade -f -s /net/server/export/s10/latest -n zfsflashBE -a /tank/data/zfs10up2flar
miniroot filesystem is <lofs>
Mounting miniroot at </net/server/s10up/latest/Solaris_10/Tools/Boot>
Validating the contents of the media </net/server/export/s10up/latest>.
The media is a standard Solaris media.
Validating the contents of the miniroot </net/server/export/s10up/latest/Solaris_10/Tools/Boot>.
Locating the flash install program.
Checking for existence of previously scheduled Live Upgrade requests.
```

BEISPIEL 4–8 Erstellen einer ZFS-BU mit einem ZFS-Flash-Archiv (`luupgrade`) *(Fortsetzung)*

```
Constructing flash profile to use.
Creating flash profile for BE <zfsflashBE>.
Performing the operating system flash install of the BE <zfsflashBE>.
CAUTION: Interrupting this process may leave the boot environment unstable or unbootable.
Extracting Flash Archive: 100% completed (of 5020.86 megabytes)
The operating system flash install completed.
updating /.alt.tmp.b-rgb.mnt/platform/sun4u/boot_archive

The Live Flash Install of the boot environment <zfsflashBE> is complete.
```

5. Aktivieren Sie die alternative BU.

```
clone-system# luactivate zfsflashBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsflashBE>.
.
.
.
Modifying boot archive service
Activation of boot environment <zfsflashBE> successful.
```

6. Starten Sie das System neu.

```
clone-system# init 6
```

Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)

Sie können ein System mit Zonen mithilfe von Live Upgrade zwar migrieren, die unterstützten Konfigurationen sind in Solaris 10 10/08 jedoch begrenzt. Nach der Installation von Solaris 10 5/09 oder einer höheren Version oder der Aktualisierung auf Solaris 10 5/09 oder eine höhere Version werden mehr Zonenkonfigurationen unterstützt. Weitere Informationen finden Sie unter „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147.

In diesem Abschnitt wird beschrieben, wie Sie ein System mit Zonen so installieren und konfigurieren müssen, dass es mit Live Upgrade aktualisiert und gepatcht werden kann. Informationen zum Migrieren eines ZFS-Root-Dateisystems ohne Zonen finden Sie unter „Migrieren oder Aktualisieren eines ZFS-Root-Dateisystems mit Live Upgrade (ohne Zonen)“ auf Seite 135.

Wenn Sie ein System mit Zonen migrieren möchten bzw. die Konfiguration eines Systems mit Zonen in Betracht ziehen, berücksichtigen•Sie in Solaris 10 10/08 die folgenden Vorgehensweisen:

- „So migrieren Sie ein UFS-Root-Dateisystem mit Zonen-Roots auf UFS auf ein ZFS-Root-Dateisystem (Solaris 10 10/08)“ auf Seite 143

- „So konfigurieren Sie ein ZFS-Root-Dateisystem mit Zonen-Roots auf ZFS (Solaris 10 10/08)“ auf Seite 145
- „So aktualisieren bzw. patchen Sie ein ZFS-Root-Dateisystem mit Zonen-Roots auf ZFS (Solaris 10 10/08)“ auf Seite 146
- „Lösen von Problemen mit ZFS-Einhängepunkten, die ein erfolgreiches Booten verhindern (Solaris 10 10/08)“ auf Seite 166

Halten Sie sich an diese empfohlenen Vorgehensweisen zum Einrichten von Zonen mit einem ZFS-Root-Dateisystem, damit Sie für dieses System Live Upgrade verwenden können.

▼ So migrieren Sie ein UFS-Root-Dateisystem mit Zonen-Roots auf UFS auf ein ZFS-Root-Dateisystem (Solaris 10 10/08)

Dieses Verfahren erklärt, wie ein UFS-Root-Dateisystem mit installierten Zonen in ein ZFS-Root-Dateisystem migriert wird, damit die ZFS-Zonen-Root-Konfiguration aktualisiert oder gepatcht werden kann.

In den nachfolgenden Schritten ist der Name des Beispiel-Pools `rpool`, und die Namen der aktiven Boot-Umgebung (BUs) beginnen mit `s10BE*`.

1 Rüsten•Sie das System auf Solaris 10 10/08 auf, falls darauf ein früheres Solaris 10-Release installiert ist.

Weitere Informationen zum Aktualisieren eines Systems, auf dem Solaris 10 installiert ist, finden Sie in *Oracle Solaris 10 1/13 Installationshandbuch: Live Upgrade und Planung von Upgrades*.

2 Erstellen Sie den Root-Pool.

```
# zpool create rpool mirror c0t1d0 c1t1d0
```

Information zu Voraussetzungen für den Root-Pool finden Sie unter „Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung“ auf Seite 113.

3 Überprüfen Sie, ob die Zonen aus der UFS-Umgebung gebootet wurden.

4 Erstellen Sie die neue ZFS-Boot-Umgebung.

```
# lucreate -n s10BE2 -p rpool
```

Dieser Befehl erstellt im Root-Pool Datasets für die neue BU und kopiert die aktuelle BU einschließlich der Zonen in diese Datasets.

5 Aktivieren Sie die neue ZFS-Boot-Umgebung.

```
# luactivate s10BE2
```

Jetzt besitzt das System ein ZFS-Root-Dateisystem, die Zonen-Roots auf UFS befinden sich jedoch noch im UFS-Root-Dateisystem. Als nächster Schritt müssen die UFS-Zonen vollständig in eine unterstützte ZFS-Konfiguration migriert werden.

6 Starten Sie das System neu.

```
# init 6
```

7 Migrieren Sie die Zonen in eine ZFS-BU.

a. Booten Sie die Zonen.

b. Erstellen Sie eine weitere ZFS-BU innerhalb des Pools.

```
# lucreate s10BE3
```

c. Aktivieren Sie die neue Boot-Umgebung.

```
# luactivate s10BE3
```

d. Starten Sie das System neu.

```
# init 6
```

Dieser Schritt stellt sicher, dass die ZFS-BU und die Zonen gebootet werden können.

8 Beseitigen Sie alle potenziellen Probleme mit Einhängepunkten.

Aufgrund eines Bugs im Live Upgrade kann das Booten der nicht aktiven BU fehlschlagen, wenn ein ZFS-Dataset oder ein ZFS-Dataset einer Zone in der BU einen ungültigen Einhängepunkt besitzt.

a. Überprüfen Sie die Ausgabe des Befehls `zfs list`.

Suchen Sie ungültige temporäre Einhängepunkte. Beispiel:

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10up

NAME                                MOUNTPOINT
rpool/ROOT/s10up                    /.alt.tmp.b-VP.mnt/
rpool/ROOT/s10up/zones              /.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10up/zones/zonerootA    /.alt.tmp.b-VP.mnt/zones/zonerootA
```

Der Einhängepunkt für die ZFS-Root-BU (`rpool/ROOT/s10up`) muss `/` sein.

b. Setzen Sie die Einhängepunkte für die ZFS-BU und ihre Datasets zurück.

Beispiel:

```
# zfs inherit -r mountpoint rpool/ROOT/s10up
# zfs set mountpoint=/ rpool/ROOT/s10up
```

c. Starten Sie das System neu.

Wählen Sie die BU aus, deren Einhängepunkte Sie gerade korrigiert haben, wenn in der Eingabeaufforderung des OpenBoot-PROM bzw. im GRUB-Menü die Option zum Booten einer spezifischen BU angezeigt wird.

▼ So konfigurieren Sie ein ZFS-Root-Dateisystem mit Zonen-Roots auf ZFS (Solaris 10 10/08)

In diesem Verfahren wird erklärt, wie ein ZFS-Root-Dateisystem und eine ZFS-Zonen-Root-Konfiguration einrichtet werden, die aktualisiert oder gepatcht werden können. In dieser Konfiguration werden die ZFS-Zonen-Roots als ZFS-Datasets erstellt.

In den nachfolgenden Schritten ist der Name des Beispiel-Pools `rpool`, und der Name der aktiven Boot-Umgebung ist `s10BE`. Der Name für das Zonen-Dataset kann ein beliebiger gültiger Dataset-Name sein. Der Name des Zonen-Dataset im folgenden Beispiel ist `zones`.

1 Installieren Sie mithilfe des interaktiven Textmodus-Installationsverfahrens oder des JumpStart-Installationsverfahrens ein System mit ZFS-Root.

Lesen Sie je nach angewendetem Installationsverfahren „Installieren eines ZFS-Root-Dateisystems (Erstinstallation von Oracle Solaris)“ auf Seite 116 oder „Installieren eines ZFS-Root-Dateisystems (JumpStart-Installation)“ auf Seite 128.

2 Booten Sie das System vom neu erstellten Root-Pool.

3 Erstellen Sie ein Dataset zum Gruppieren der Zonen-Roots.

Beispiel:

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones
```

Das Setzen des Wertes `noauto` für die Eigenschaft `canmount` garantiert, dass das Dataset nur von der speziellen Aktion von Live Upgrade und dem Startup-Code des Systems eingehängt werden kann.

4 Hängen Sie das neu erstellte Zonen-Dataset ein.

```
# zfs mount rpool/ROOT/s10BE/zones
```

Das Dataset wird unter `/zones` eingehängt.

5 Erstellen Sie für jede Zonen-Root ein•Dataset und hängen Sie dieses ein.

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones/zonerootA
# zfs mount rpool/ROOT/s10BE/zones/zonerootA
```

6 Setzen Sie für das Zonen-Root-Verzeichnis die entsprechenden Zugriffsrechte.

```
# chmod 700 /zones/zonerootA
```

7 Konfigurieren Sie die Zone und setzen Sie den Zonen-Pfad wie folgt:

```
# 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
```

Mithilfe der folgenden Syntax können Sie festlegen, dass die Zonen beim Booten des Systems automatisch gebootet werden sollen:

```
zonecfg:zoneA> set autoboot=true
```

8 Installieren Sie die Zone.

```
# zoneadm -z zoneA install
```

9 Booten Sie die Zone.

```
# zoneadm -z zoneA boot
```

▼ So aktualisieren bzw. patchen Sie ein ZFS-Root-Dateisystem mit Zonen-Roots auf ZFS (Solaris 10 10/08)

Verwenden Sie dieses Verfahren, wenn Sie ein ZFS-Root-Dateisystem mit Zonen-Roots auf ZFS aktualisieren bzw. patchen müssen. Solche Aktualisierungen umfassen entweder eine Systemaktualisierung oder die Anwendung von Patches.

In den nachfolgenden Schritten ist `newBE` der Beispielname der BU, die aktualisiert bzw. gepatcht werden soll.

1 Erstellen Sie die BU, die aktualisiert oder gepatcht werden soll.

```
# lucreate -n newBE
```

Die vorhandene BU sowie sämtliche Zonen werden geklont. Für jedes Dataset der ursprünglichen BU wird ein Dataset erstellt. Die neuen Datasets werden im gleichen Pool erstellt, in dem sich auch der aktuelle Root-Pool befindet.

2 Wählen Sie eines der folgenden Verfahren aus, um das System zu aktualisieren oder Patches auf die neue BU anzuwenden:

- Rüsten Sie das System auf.

  ```
  # luupgrade -u -n newBE -s /net/install/export/s10up/latest
  ```

  Die Option `-s` gibt an, wo sich der Oracle Solaris-Installationsdatenträger befindet.

- Wenden Sie Patches auf die neue BU an.

  ```
  # luupgrade -t -n newBE -t -s /patchdir 139147-02 157347-14
  ```

3. **Aktivieren Sie die neue BU.**

```
# luactivate newBE
```

4. **Booten Sie von der neu aktivierten BU.**

```
# init 6
```

5. **Beseitigen Sie alle potenziellen Probleme mit Einhängepunkten.**

   Aufgrund eines Bugs im Live Upgrade kann das Booten der nicht aktiven BU fehlschlagen, wenn ein ZFS-Dataset oder ein ZFS-Dataset einer Zone in der BU einen ungültigen Einhängepunkt besitzt.

   a. **Überprüfen Sie die Ausgabe des Befehls `zfs list`.**

      Suchen Sie ungültige temporäre Einhängepunkte. Beispiel:

```
# 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
```

      Der Einhängepunkt für die ZFS-Root-BU (`rpool/ROOT/newBE`) muss `/` sein.

   b. **Setzen Sie die Einhängepunkte für die ZFS-BU und ihre Datasets zurück.**

      Beispiel:

```
# zfs inherit -r mountpoint rpool/ROOT/newBE
# zfs set mountpoint=/ rpool/ROOT/newBE
```

   c. **Starten Sie das System neu.**

      Wählen Sie die Boot-Umgebung aus, deren Einhängepunkte Sie gerade korrigiert haben, wenn im GRUB-Menü bzw. in der Eingabeaufforderung des OpenBoot-PROM die Option zum Booten einer spezifischen Boot-Umgebung angezeigt wird.

Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)

Ab Solaris 10 10/08 können Sie Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen verwenden. Zusätzliche Sparse-Zonenkonfigurationen (Root- und Gesamtzonen) werden durch Live Upgrade ab Solaris 10 5/09 unterstützt.

In diesem Abschnitt wird beschrieben, wie Sie ein System mit Zonen konfigurieren, damit es mit Live Upgrade ab Solaris 10 5/09 aktualisiert und gepatcht werden kann. Informationen zum

Migrieren eines ZFS-Root-Dateisystems ohne Zonen finden Sie unter „Migrieren oder Aktualisieren eines ZFS-Root-Dateisystems mit Live Upgrade (ohne Zonen)“ auf Seite 135.

Berücksichtigen Sie folgende Punkte, wenn Sie Oracle Solaris Live Upgrade für ZFS und Zonen ab Solaris 10 5/09 verwenden:

- Wenn Sie Live Upgrade für in Solaris 10 5/09 unterstützte Zonenkonfigurationen verwenden, müssen Sie zunächst mit dem standardmäßigen Upgrade-Programm eine Aktualisierung auf Solaris 10 5/09 oder eine höhere Version durchführen.
- Dann können Sie mit Live Upgrade Ihr UFS-Root-Dateisystem mit Zonen-Roots in ein ZFS-Root-Dateisystem migrieren oder ein Upgrade oder Patch auf Ihr ZFS-Root-Dateisystem und Zonen-Roots anwenden.
- Nicht unterstützte Zonenkonfigurationen aus einer früheren Solaris 10-Version können nicht direkt in Solaris 10 5/09 migriert werden.

Anhand der folgende Vorgehensweisen können Sie mit Solaris ab Version 10 5/09 ein System mit Zonen migrieren oder konfigurieren:

- „Unterstütztes ZFS mit Zonen-Root-Konfigurationsinformationen (ab Solaris 10 5/09)“ auf Seite 148
- „So erstellen Sie eine ZFS-BU mit einem ZFS-Root-Dateisystem und einem Zonen-Root (ab Solaris 10 5/09)“ auf Seite 150
- „So aktualisieren oder patchen Sie ein ZFS-Root-Dateisystem mit Zonen-Roots (ab Solaris 10 5/09)“ auf Seite 152
- „So migrieren Sie ein UFS-Root-Dateisystem mit Zonen-Roots in ein ZFS-Root-Dateisystem (ab Solaris 10 5/09)“ auf Seite 155

Unterstütztes ZFS mit Zonen-Root-Konfigurationsinformationen (ab Solaris 10 5/09)

Überprüfen Sie die unterstützten Zonenkonfigurationen, bevor Sie Oracle Solaris Live Upgrade zur Migration oder zum Upgrade eines Systems mit Zonen verwenden.

- **Migrieren eines UFS-Root-Dateisystems in ein ZFS-Root-Dateisystem** – Die folgenden Zonen-Root-Konfigurationen werden unterstützt:
  - In einem Verzeichnis des UFS-Root-Dateisystems
  - In einem Unterverzeichnis eines Einhängepunkts im UFS-Root-Dateisystem
  - Ein UFS-Root-Dateisystem mit einem Zonen-Root in einem UFS-Root-Dateisystem-Verzeichnis eines UFS-Root-Dateisystem-Einhängepunkts und ein ZFS-Nicht-Root-Pool mit einem Zonen-Root

  Ein UFS-Root-Dateisystem mit einem Zonen-Root als Einhängepunkt wird nicht unterstützt.

- **Migrieren oder Aktualisieren eines UFS-Root-Dateisystems auf ein ZFS-Root-Dateisystem** – Die folgenden Zonen-Root-Konfigurationen werden unterstützt:
  - In einem Dateisystem in einem ZFS-Root oder einem Pool außerhalb des Root. `/zonepool/zones` ist beispielsweise zulässig. In einigen Fällen erstellt Live Upgrade ein Dateisystem für den Zonen-Root (`zoneds`), wenn vor dem Live Upgrade kein solches Dateisystem vorhanden ist.
  - In einem untergeordneten Dateisystem oder Unterverzeichnis eines ZFS-Dateisystems, solange keine anderen Zonenpfade verschachtelt sind. `/zonepool/zones/zone1` und `/zonepool/zones/zone1_dir` sind beispielsweise zulässig.

    Im folgenden Beispiel ist `zonepool/zones` ein Dateisystem, das die Zonen-Roots enthält, und `rpool` enthält die ZFS-BU:

    ```
    zonepool
    zonepool/zones
    zonepool/zones/myzone
    rpool
    rpool/ROOT
    rpool/ROOT/myBE
    ```

    Mit der folgenden Syntax erstellt Live Upgrade Schnappschüsse und Klone der Zonen in `zonepool` und der BU `rpool`:

    ```
    # lucreate -n newBE
    ```

    Die BU `newBE` in `rpool/ROOT/newBE` wird erstellt. Wenn die BU `newBE` aktiviert ist, bietet sie Zugriff auf die `zonepool`-Komponenten.

    Wäre `/zonepool/zones` im vorherigen Beispiel ein Unterverzeichnis und kein separates Dateisystem, dann würde Live Upgrade es als Komponente des Root-Pools `rpool` migrieren.
  - **Die folgende ZFS- und Zonenkonfiguration wird nicht unterstützt:**
    - Live Upgrade kann nicht zur Erstellung einer alternativen BU verwendet werden, wenn die Quell-BU über eine nicht globale Zone mit einem Zonenpfad verfügt, der auf den Einhängepunkt eines Pool-Dateisystems der obersten Hierarchieebene gesetzt ist. Wenn beispielsweise der Pool `zonepool` über ein als `/zonepool` eingehängtes Dateisystem verfügt, darf keine nicht globale Zone mit einem auf `/zonepool` gesetzten Zonenpfad vorhanden sein.
    - Fügen Sie keinen Dateisystemeintrag für eine nicht-globale Zone in der Datei `/etc/vfstab` der globalen Zone hinzu. Verwenden Sie stattdessen die Funktion `add fs` von `zonecfg`, um ein Dateisystem zu einer nicht-globalen Zone hinzuzufügen.
- **Informationen zur Migration oder Aktualisierung von UFS- und ZFS-Zonen** – Berücksichtigen Sie folgende Informationen zur Migration oder Aktualisierung einer UFS- oder ZFS-Umgebung:

- Wenn Sie Ihre Zonen wie unter „Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)“ auf Seite 142 in Solaris 10 10/08 konfiguriert und eine Aktualisierung auf Solaris 10 5/09 oder eine höhere Version durchgeführt haben, können Sie in ein ZFS-Root-Dateisystem migrieren oder Solaris Live Upgrade für eine Aktualisierung auf Solaris 10 5/09 oder eine höhere Version verwenden.
- Erstellen Sie keine Zonen-Roots in verschachtelten Verzeichnissen wie `zones/zone1` und `zones/zone1/zone2`. Anderenfalls kann das Einhängen während des Boot-Vorgangs fehlschlagen.

▼ So erstellen Sie eine ZFS-BU mit einem ZFS-Root-Dateisystem und einem Zonen-Root (ab Solaris 10 5/09)

Verwenden Sie dieses Verfahren, nachdem Sie eine Erstinstallation von Solaris 10 5/09 oder einer höheren Version durchgeführt haben, um ein ZFS-Root-Dateisystem zu erstellen. Verwenden Sie dieses Verfahren ebenfalls, nachdem Sie den Befehl `luupgrade` verwendet haben, um ein ZFS-Root-Dateisystem auf Solaris 10 5/09 oder eine höhere Version zu aktualisieren. Eine mit dieser Vorgehensweise erstellte ZFS-BU kann aktualisiert oder gepatcht werden.

Das in den folgenden Schritten verwendete Oracle Solaris 10 9/10-Beispielsystem besitzt ein ZFS-Root-Dateisystem und ein Zonen-Root-Dataset in `/rpool/zones`. Eine ZFS-BU mit dem Namen `zfs2BE` wird erstellt, die aktualisiert oder gepatcht werden kann.

1 Überprüfen Sie die vorhandenen ZFS-Dateisysteme.

```
# 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 Stellen Sie sicher, dass die Zonen installiert und gebootet sind.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /rpool/zones                   native   shared
```

3 Erstellen Sie die ZFS-BU.

```
# 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 Aktivieren Sie die ZFS-BU.

```
# 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>.
.
.
.
```

5 Booten Sie die ZFS-BU.

```
# init 6
```

6 Überprüfen Sie, ob die ZFS-Dateisysteme und Zonen in der neuen BU erstellt wurden.

```
# 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
```

▼ So aktualisieren oder patchen Sie ein ZFS-Root-Dateisystem mit Zonen-Roots (ab Solaris 10 5/09)

Verwenden Sie dieses Verfahren, wenn Sie ein ZFS-Root-Dateisystem mit Zonen-Roots in Solaris 10 5/09 oder einer höheren Version aktualisieren oder patchen müssen. Solche Aktualisierungen umfassen entweder eine Systemaktualisierung oder die Anwendung von Patches.

In den nachfolgenden Schritten ist `zfs2BE` der Beispielname der BU, die aktualisiert bzw. gepatcht werden soll.

1 Überprüfen Sie die vorhandenen ZFS-Dateisysteme.

```
# 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 Stellen Sie sicher, dass die Zonen installiert und gebootet sind.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   5 zfszone          running    /rpool/zones                   native   shared
```

3 Erstellen Sie die ZFS-BU, die aktualisiert oder gepatcht werden soll.

```
# 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. **Wählen Sie eines der folgenden Verfahren aus, um das System zu aktualisieren oder Patches auf die neue BU anzuwenden:**

   - Rüsten Sie das System auf.

     ```
     # luupgrade -u -n zfs2BE -s /net/install/export/s10up/latest
     ```

     Die Option `-s` gibt an, wo sich der Oracle Solaris-Installationsdatenträger befindet.

     Dieser Vorgang kann sehr viel Zeit beanspruchen.

     Ein vollständiges Beispiel für den `luupgrade`-Vorgang finden Sie in Beispiel 4–9.

   - Wenden Sie Patches auf die neue BU an.

     ```
     # luupgrade -t -n zfs2BE -t -s /patchdir patch-id-02 patch-id-04
     ```

5. **Aktivieren Sie die neue Boot-Umgebung.**

```
# 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. **Booten Sie das System von der neu aktivierten Boot-Umgebung.**

   ```
   # init 6
   ```

Beispiel 4–9 Aktualisieren eines ZFS-Root-Dateisystems mit Zonen-Root auf ein ZFS-Root-Dateisystem in Oracle Solaris 10 9/10

In diesem Beispiel wird eine auf einem Solaris 10 /10/09-System erstellte ZFS-BU (`zfsBE`) mit einem ZFS-Root-Dateisystem und einer Zonen-Root in einem Pool außerhalb des Root auf Oracle Solaris 10 9/10 aktualisiert. Dieser Vorgang kann viel Zeit beanspruchen. Dann wird die aktualisierte BU (`zfs2BE`) aktiviert. Stellen Sie vor dem Aktualisierungsversuch sicher, dass die Zonen installiert und gebootet wurden.

In diesem Beispiel werden der Pool `zonepool`, das Dataset `/zonepool/zones` und die Zone `zfszone` wie folgt erstellt:

```
# 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 zfs2BE
# 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
```

So migrieren Sie ein UFS-Root-Dateisystem mit Zonen-Roots in ein ZFS-Root-Dateisystem (ab Solaris 10 5/09)

Verwenden Sie dieses Verfahren, um ein System mit einem UFS-Root-Dateisystem und einer Zonen-Root auf Solaris 10 5/09 oder eine höhere Version zu migrieren. Erstellen Sie dann mit Live Upgrade eine ZFS-BU.

In den folgenden Schritten lautet der Name der UFS-Beispiel-BU `c1t1d0s0`, der UFS-Zonen-Root heißt `zonepool/zfszone` und die ZFS-Root-BU heißt `zfsBE`.

1 Aktualisieren Sie das System auf Solaris 10 5/09 oder eine höhere Version, falls darauf eine frühere Solaris 10-Version installiert ist.

Weitere Informationen zum Aktualisieren eines Systems, auf dem Solaris 10 installiert ist, finden Sie in *Oracle Solaris 10 1/13 Installationshandbuch: Live Upgrade und Planung von Upgrades*.

2 Erstellen Sie den Root-Pool.

Information zu Voraussetzungen für den Root-Pool finden Sie unter „Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung“ auf Seite 113.

3 Überprüfen Sie, ob die Zonen aus der UFS-Umgebung gebootet wurden.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /zonepool/zones                native   shared
```

4 Erstellen Sie die neue ZFS-BU.

```
# lucreate -c c1t1d0s0 -n zfsBE -p rpool
```

Dieser Befehl erstellt im Root-Pool Datasets für die neue BU und kopiert die aktuelle BU einschließlich der Zonen in diese Datasets.

5 Aktivieren Sie die neue ZFS-BU.

```
# 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 Starten Sie das System neu.

```
# init 6
```

7 Überprüfen Sie, ob die ZFS-Dateisysteme und Zonen in der neuen BU erstellt wurden.

```
# 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
```

Beispiel 4–10 Migrieren eines UFS-Root-Dateisystems mit Zonen-Root in ein ZFS-Root-Dateisystem

In diesem Beispiel wird ein Oracle Solaris 10 9/10-System mit einem UFS-Root-Dateisystem, einem Zonen-Root (`/uzone/ufszone`), einem ZFS-Pool außerhalb des Roots (`pool`) und einem Zonen-Root (`/pool/zfszone`) in ein ZFS-Root-Dateisystem migriert. Stellen Sie vor dem Migrationsversuch sicher, dass der ZFS-Root-Pool erstellt sowie die Zonen installiert und gebootet wurden.

```
# 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
```

Verwalten der ZFS-Swap- und Dump-Geräte

Während einer Erstinstallation des Betriebssystems Oracle Solaris oder nach der Durchführung einer Live Upgrade-Migration aus einem UFS-Dateisystem wird auf einem ZFS-Volume im ZFS-Root-Pool ein Swap-Bereich erstellt. Beispiel:

```
# swap -l
swapfile                  dev  swaplo  blocks   free
/dev/zvol/dsk/rpool/swap 256,1      16 4194288 4194288
```

Während einer Erstinstallation des Betriebssystems Oracle Solaris oder eines Live Upgrade in einem UFS-Dateisystem wird auf einem ZFS-Volume im ZFS-Root-Pool ein Dump-Gerät erstellt. Im Allgemeinen ist keine Administration des Dump-Geräts erforderlich, da es während der Installation automatisch eingerichtet wird. Beispiel:

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
  Savecore enabled: yes
   Save compressed: on
```

Wenn Sie das Dump-Gerät deaktivieren und entfernen, müssen Sie es nach der Neuerstellung mit dem Befehl `dumpadm` aktivieren. In den meisten Fällen werden Sie nur die Größe des Dump-Geräts anpassen müssen. Dazu verwenden Sie den Befehl `zfs`.

Informationen zu den von den Installationsprogrammen erstellten Swap- und Dump-Volume-Größen finden Sie unter „Oracle Solaris-Installation und Live Upgrade: Voraussetzungen für die ZFS-Unterstützung“ auf Seite 113.

Die Swap- und Dump-Volume-Größe kann während und nach der Installation geändert werden. Weitere Informationen finden Sie unter „Anpassen der Größe von ZFS-Swap- und Dump-Geräten“ auf Seite 159.

Berücksichtigen Sie bei der Arbeit mit ZFS-Swap- und Dump-Geräten die folgenden Probleme:

- Für Swap-Bereich und Dump-Gerät müssen separate ZFS-Volumes verwendet werden.
- Derzeit wird die Verwendung von Swap-Dateien in ZFS-Dateisystemen nicht unterstützt.
- Wenn Sie den Swap-Bereich oder das Dump-Gerät nach der Installation oder dem Upgrade des Systems ändern müssen, benutzen Sie hierzu die Befehle `swap` und `dumpadm` wie in vorherigen Versionen. Weitere Informationen finden Sie in Kapitel 16, „Configuring

Additional Swap Space (Tasks)“ in *System Administration Guide: Devices and File Systems* und Kapitel 17, „Managing System Crash Information (Tasks)“ in *System Administration Guide: Advanced Administration*.

Weitere Informationen finden Sie in den folgenden Abschnitten:

- „Anpassen der Größe von ZFS-Swap- und Dump-Geräten“ auf Seite 159
- „Behebung von Problemen mit ZFS-Dump-Geräten“ auf Seite 161

Anpassen der Größe von ZFS-Swap- und Dump-Geräten

Nach der Installation müssen Sie möglicherweise die Größe der Swap- und Dump-Geräte anpassen oder die Swap- und Dump-Volumes neu erstellen.

- Die Größe der Swap- und Dump-Volumes kann während einer Erstinstallation angepasst werden. Weitere Informationen finden Sie in Beispiel 4–1.
- Sie können vor Ausführung von Live Upgrade Swap- und Dump-Volumes erstellen und deren Größe festlegen. Beispiel:

  1. Erstellen Sie den Speicher-Pool.

     ```
     # zpool create rpool mirror c0t0d0s0 c0t1d0s0
     ```

  2. Erstellen Sie das Dump-Gerät.

     ```
     # zfs create -V 2G rpool/dump
     ```

  3. Aktivieren Sie das Dump-Gerät.

     ```
     # 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. Erstellen Sie das Swap-Volume:

     ```
     # zfs create -V 2G rpool/swap
     ```

  5. Wenn ein Swap-Gerät hinzugefügt oder verändert wird, müssen Sie den Swap-Bereich aktivieren.

     ```
     # swap -a /dev/zvol/dsk/rpool/swap
     ```

  6. Fügen Sie einen Eintrag für das Swap-Volume in die Datei `/etc/vfstab` ein.

  Live Upgrade bietet keine Möglichkeit, die Größe vorhandener Swap- und Dump-Volumes zu ändern.

- Sie können nach der Installation des Systems die Eigenschaft `volsize` des Dump-Geräts zurücksetzen. Beispiel:

```
# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME        PROPERTY  VALUE       SOURCE
rpool/dump  volsize   2G          -
```

- Wenn der aktuelle Swap-Bereich nicht verwendet wird, können Sie die Größe des aktuellen Swap-Volumes skalieren, müssen jedoch das System neu starten, um die erhöhte Swap-Kapazität zu sehen.

  ```
  # zfs get volsize rpool/swap
  NAME        PROPERTY  VALUE    SOURCE
  rpool/swap  volsize   4G       local
  # zfs set volsize=8g rpool/swap
  # zfs get volsize rpool/swap
  NAME        PROPERTY  VALUE    SOURCE
  rpool/swap  volsize   8G       local
  # init 6
  ```

- Sie können versuchen, das Swap-Volume zu skalieren, am besten entfernen Sie jedoch das Swap-Gerät. Erstellen Sie es anschließend erneut. Beispiel:

  ```
  # swap -d /dev/zvol/dsk/rpool/swap
  # zfs create -V 2g rpool/swap
  # swap -a /dev/zvol/dsk/rpool/swap
  ```

- Mit folgender Profilsyntax lässt sich die Größe der Swap- und Dump-Volumes in einem JumpStart-Profil anpassen:

  ```
  install_type initial_install
  cluster SUNWCXall
  pool rpool 16g 2g 2g c0t0d0s0
  ```

  In diesem Profil werden die Größe des Swap-Volume und des Dump-Volume mit zwei `2g`-Einträgen auf je 2 GB festgelegt.

- Wenn Sie mehr Swap-Speicherplatz auf einem bereits installierten System benötigen, fügen Sie einfach ein weiteres Swap-Volume hinzu. Beispiel:

  ```
  # zfs create -V 2G rpool/swap2
  ```

  Aktivieren Sie dann das neue Swap-Volume. Beispiel:

  ```
  # 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
  ```

  Fügen Sie als Letztes einen zweiten Eintrag für das Swap-Volume in die Datei `/etc/vfstab` ein.

Anpassen der ZFS-Swap- und Dump-Volumes

Beachten Sie Folgendes, wenn Sie die Standard-Swap- und -Dump-Volumes entfernen und in einem Nicht-Root-Pool (Datenpool) neu erstellen:

- Wenn Sie Swap- und Dump-Geräte in einem Nicht-Root-Pool erstellen möchten, erstellen Sie keine Swap- und Dump-Volumes in einem RAIDZ-Pool. Wenn ein Pool Swap- und Dump-Volumes umfasst, muss es sich um einen Pool auf einer einzigen Festplatte oder um einen gespiegelten Pool handeln.
- Wenn Sie Live Upgrade zur Aktualisierung Ihres Systems verwenden, können Sie das Dump-Gerät mit der Option `-P` von PBU nach ABU verschieben und somit beibehalten. Beispiel:

  ```
  # lucreate -n newBE -P
  ```

Behebung von Problemen mit ZFS-Dump-Geräten

Überprüfen Sie Folgendes, wenn Sie Probleme mit der Erfassung eines Systemabsturz-Speicherabzugs oder dem Ändern der Größe des Dump-Geräts haben.

- Wurde bei einem Systemabsturz nicht automatisch ein Speicherabzug erstellt, können Sie den Befehl `savecore` verwenden, um den Speicherabzug zu speichern.
- Wenn Sie erstmals ein ZFS-Root-Dateisystem erstellen oder in ein ZFS-Root-Dateisystem migrieren, wird automatisch ein Dump-Volume erstellt. In den meisten Fällen müssen Sie nur die Größe des Dump-Volume anpassen, wenn die Größe des Standard-Dump-Volume zu klein ist. Beispielsweise wird die Größe des Dump-Volumes in einem System mit hoher Speicherkapazität wie folgt auf 40 GB erhöht:

  ```
  # zfs set volsize=40G rpool/dump
  ```

  Das Ändern der Größe eine großen Dump-Volumes kann sehr zeitaufwändig sein.

  Wenn Sie aus irgendeinem Grund ein Dump-Gerät aktivieren müssen, nachdem Sie manuell ein Dump-Gerät erstellt haben, verwenden Sie folgende Syntax:

  ```
  # 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
  ```

- Ein System mit einer Speicherkapazität von 128 GB oder mehr benötigt möglicherweise ein größeres Dump-Gerät als jenes, das standardmäßig erstellt wurde. Wenn das Dump-Gerät zu klein ist, um bei einem Systemabsturz einen Speicherabzug zu erfassen, wird eine Meldung wie die folgende angezeigt:

  ```
  # 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
```

Informationen zum Ändern der Größe von Swap- und Dump-Geräten finden Sie unter „Planning for Swap Space“ in *System Administration Guide: Devices and File Systems*.

- Sie können derzeit kein Dump-Gerät zu einem Pool mit mehreren Geräten der obersten Hierarchieebene hinzufügen. Eine Meldung ähnlich der Folgenden wird angezeigt:

```
# dumpadm -d /dev/zvol/dsk/datapool/dump
dump is not supported on device '/dev/zvol/dsk/datapool/dump': 'datapool' has multiple top level vdevs
```

Fügen Sie das Dump-Gerät zum Root-Pool hinzu, der nicht mehrere Geräte der obersten Ebene enthalten kann.

Booten aus einem ZFS-Root-Dateisystem

Sowohl bei SPARC- als auch bei x86-Systemen kommt die neue Art des Bootens mit Boot-Archiv zum Einsatz. Dabei handelt es sich um ein Dateisystemabbild, in dem sich die zum Booten benötigten Dateien befinden. Beim Booten aus einem ZFS-Root-Dateisystem werden die Pfadnamen des Archivs und der Kernel-Datei in dem für das Booten ausgewählten Root-Dateisystem aufgelöst.

Wenn das System für die Installation gebootet wird, wird während des gesamten Installationsvorgangs ein RAM-Datenträger für das Root-Dateisystem verwendet.

Das Booten aus einem ZFS-Dateisystem unterscheidet sich vom Booten aus einem UFS-Dateisystem dadurch, dass ein Gerätebezeichner bei ZFS kein einzelnes Root-Dateisystem sondern einen Speicher-Pool kennzeichnet. Ein Speicher-Pool kann mehrere *bootfähige Datasets* oder ZFS-Root-Dateisysteme enthalten. Beim Booten aus ZFS müssen Sie ein Boot-Gerät und ein Root-Dateisystem innerhalb des Pools angeben, das durch das Boot-Gerät identifiziert ist.

Standardmäßig ist das zum Booten ausgewählte Dataset das mit der Eigenschaft `bootfs` des Pools bezeichnete Dataset. Diese Standardauswahl kann durch Angabe eines alternativen bootfähigen Datasets mit dem Befehl `boot -Z` außer Kraft gesetzt werden.

Booten von einer alternativen Festplatte in einem gespiegelten ZFS-Root-Pool

Bei der Installation des Systems haben Sie die Möglichkeit, einen ZFS-Root-Pool mit Datenspiegelung zu erstellen oder eine Festplatte einzubinden, um nach der Installation ein solches zu erstellen. Weitere Informationen finden Sie unter:

- „Installieren eines ZFS-Root-Dateisystems (Erstinstallation von Oracle Solaris)“ auf Seite 116
- „Erstellen eines gespiegelten ZFS-Root-Pools (nach der Installation)“ auf Seite 122

Beachten Sie die folgenden bekannten Probleme im Zusammenhang mit gespiegelten ZFS-Root-Pools:

- Wenn Sie mit dem Befehl `zpool replace` eine Root-Pool-Festplatte ersetzen, müssen Sie mithilfe des Befehls `installboot` oder `installgrub` die Boot-Informationen auf der neu ersetzten Festplatte installieren. Dieser Schritt ist nicht notwendig, wenn Sie zur Erstellung des gespiegelten ZFS-Root-Pools die Erstinstallationsmethode verwenden oder mit dem Befehl `zpool attach` eine Festplatte in den Root-Pool einbinden. Die Syntax der Befehle `installboot` und `installgrub` lautet wie folgt:
  - SPARC:

    ```
    sparc# installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk
    ```

  - x86:

    ```
    x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c0t1d0s0
    ```

- Es kann von verschiedenen Speichergeräten in einem gespiegelten ZFS-Root-Pool gebootet werden. Je nach der Hardwarekonfiguration kann es erforderlich sein, eine Aktualisierung von PROM oder BIOS vorzunehmen, um ein anderes Boot-Gerät angeben zu können.

  So ist es in diesem Pool beispielsweise möglich, vom Datenträger (`c1t0d0s0` oder `c1t1d0s0`) zu booten:

  ```
  # zpool status rpool
    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: Geben Sie an der Eingabeaufforderung `ok` die alternative Festplatte an. Beispiel:

  ```
  ok boot /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
  ```

  Rufen Sie nach dem Neustart des Systems eine Bestätigung des aktiven Boot-Geräts ab. Beispiel:

  ```
  SPARC# prtconf -vp | grep bootpath
          bootpath:  ’/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a’
  ```

- x86: Wählen Sie aus dem entsprechenden BIOS-Menü eine alternative Festplatte im gespiegelten ZFS-Root-Pool aus.

  Anschließend verwenden Sie folgende Syntax, um zu bestätigen, dass Sie von der alternativen Festplatte gebootet haben.

```
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: Booten aus einem ZFS-Root-Dateisystem

Auf SPARC-Systemen mit mehreren ZFS-BUs können Sie mithilfe des Befehls `luactivate` von jeder BU booten.

Während der Installation des Betriebssystems Oracle Solaris und des Live Upgrade wird dem standardmäßigen ZFS-Root-Dateisystem automatisch die Eigenschaft `bootfs` zugewiesen.

Innerhalb eines Pools können mehrere bootfähige Datasets vorhanden sein. Standardmäßig bezieht sich die Eigenschaft `bootfs` des Pools auf den Eintrag für das bootfähige Dataset in der Datei */Pool-Name*`/boot/menu.lst` . Der Eintrag `menu.lst` kann jedoch den Befehl `bootfs` enthalten, der ein alternatives Dataset im Pool angibt. So ist es möglich, dass die Datei `menu.lst` Einträge für mehrere Root-Dateisysteme im Pool enthält.

Wenn auf einem System ein ZFS-Root-Dateisystem installiert oder eine Migration in ein ZFS-Root-Dateisystem vorgenommen wird, wird die Datei `menu.lst` um einen Eintrag wie diesen erweitert:

```
title zfsBE
bootfs rpool/ROOT/zfsBE
title zfs2BE
bootfs rpool/ROOT/zfs2BE
```

Bei Erstellung einer neuen BU wird die Datei `menu.lst` automatisch aktualisiert.

Auf SPARC-Systemen stehen zwei ZFS-Boot-Optionen zur Verfügung:

- Nach der Aktivierung der BU können Sie mit dem Befehl boot `-L` eine Liste der bootfähigen Datasets innerhalb eines ZFS-Pools anzeigen lassen. Anschließend können Sie eines der bootfähigen Datasets in der Liste auswählen. Es werden ausführliche Anweisungen zum Booten dieses Datasets angezeigt. Befolgen Sie diese Anweisungen zum Booten des ausgewählten Datasets.
- Zum Booten eines bestimmten ZFS-Datasets können Sie den Befehl boot `-Z` *dataset* verwenden.

BEISPIEL 4–11 SPARC: Booten aus einer bestimmten ZFS-Boot-Umgebung

Wenn in einem ZFS-Speicher-Pool auf dem Boot-Gerät des Systems mehrere ZFS-BUs vorhanden sind, können Sie mit dem Befehl `luactivate` eine Standard-BU angeben.

Die folgende Ausgabe von `lustatus` zeigt beispielsweise, dass zwei ZFS-BUs verfügbar sind:

BEISPIEL 4–11 SPARC: Booten aus einer bestimmten ZFS-Boot-Umgebung *(Fortsetzung)*

```
# 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     -
```

Sind auf einem SPARC-System mehrere ZFS-BUs vorhanden, können Sie den Befehl `boot -L` verwenden, um aus einer BU zu booten, die sich von der Standard-BU unterscheidet. Jedoch wird eine BU, die aus einer `boot -L`-Sitzung gebootet wird, nicht auf die Standard-BU zurückgesetzt, und die Eigenschaft `bootfs` wird nicht aktualisiert. Wenn Sie die BU, die aus einer `boot -L` Sitzung gebootet wurde, zur Standard-BU machen möchten, müssen Sie sie mit dem Befehl `luactivate` aktivieren.

Beispiel:

```
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
```

BEISPIEL 4–12 SPARC: Booten eines ZFS-Dateisystems im Failsafe-Modus

Auf SPARC-Systemen kann aus dem in `/platform/‘uname -i‘/failsafe` befindlichen Failsafe-Archiv gebootet werden. Gehen Sie dazu wie folgt vor:

```
ok boot -F failsafe
```

Um ein Failsafe-Archiv aus einem bestimmten bootfähigen ZFS-Dataset zu booten, verwenden Sie folgende Syntax:

```
ok boot -Z rpool/ROOT/zfsBE -F failsafe
```

x86: Booten aus einem ZFS-Root-Dateisystem

Während der Installation des Betriebssystems Oracle Solaris oder des Live Upgrade werden der Datei */pool-name*`/boot/grub/menu.lst` die folgenden Einträge hinzugefügt, damit ZFS automatisch gebootet wird:

```
title Solaris 10 1/13  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
```

Enthält das von GRUB als Boot-Gerät identifizierte Speichergerät einen ZFS-Speicher-Pool, wird auf Grundlage der Datei `menu.lst` das GRUB-Menü erstellt.

Auf x86-Systemen mit mehreren ZFS-BUs kann die BU über das GRUB-Menü gewählt werden. Ist das diesem Menüeintrag entsprechende Root-Dateisystem ein ZFS-Dataset, wird die folgende Option hinzugefügt:

```
-B $ZFS-BOOTFS
```

BEISPIEL 4–13 x86: Booten eines ZFS-Dateisystems

Beim Booten eines Systems von einem ZFS-Dateisystem wird das Root-Gerät von dem Boot-Parameter `-B $ZFS-BOOTFS` angegeben. Beispiel:

```
title Solaris 10 1/13  X86
findroot (pool_rpool,0,a)
kernel /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

BEISPIEL 4–14 x86: Booten eines ZFS-Dateisystems im Failsafe-Modus

Das Failsafe-Archiv für x86-Systeme lautet `/boot/x86.miniroot-safe` und kann durch Auswahl des Solaris-Failsafe-Eintrags im GRUB-Menü gebootet werden. Beispiel:

```
title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Lösen von Problemen mit ZFS-Einhängepunkten, die ein erfolgreiches Booten verhindern (Solaris 10 10/08)

Die beste Methode zum Ändern der aktiven Boot-Umgebung (BU) ist der Befehl `luactivate`. Wenn das Booten von der aktiven BU wegen eines ungültigen Patches oder eines Konfigurationsfehlers fehlschlägt, können Sie nur von einer anderen BU booten, wenn Sie diese

zur Boot-Zeit auswählen. Sie können eine alternative BU durch explizites Booten vom PROM oder von einem SPARC-System oder aus dem GRUB-Menü auf einem x86-System auswählen.

Aufgrund eines Bugs im Live Upgrade von Solaris 10 10/08 kann das Booten der nicht aktiven BU fehlschlagen, wenn ein ZFS-Dataset oder ein ZFS-Dataset einer Zone in der BU einen ungültigen Einhängepunkt besitzt. Der gleiche Bug verhindert auch das Einhängen einer BU, wenn sie ein separates `/var`-Dataset besitzt.

Wenn ein ZFS-Dataset einer Zone einen ungültigen Einhängepunkt besitzt, kann dieser mit den folgenden Schritten korrigiert werden.

▼ So lösen Sie Probleme mit ZFS-Einhängepunkten

1 Booten Sie das System von einem Failsafe-Archiv.

2 Importieren Sie den Pool.

Beispiel:

```
# zpool import rpool
```

3 Suchen Sie ungültige temporäre Einhängepunkte.

Beispiel:

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10up

    NAME                              MOUNTPOINT
    rpool/ROOT/s10up                  /.alt.tmp.b-VP.mnt/
    rpool/ROOT/s10up/zones            /.alt.tmp.b-VP.mnt//zones
    rpool/ROOT/s10up/zones/zonerootA  /.alt.tmp.b-VP.mnt/zones/zonerootA
```

Der Einhängepunkt für die Root-BU (`rpool/ROOT/s10up`) muss `/` sein.

Wenn der Boot-Vorgang wegen Einhängeproblemen von `/var` fehlschlägt, sollten Sie einen ähnlichen unzulässigen Einhängepunkt für das `/var`-Dataset suchen.

4 Setzen Sie die Einhängepunkte für die ZFS-BU und ihre Datasets zurück.

Beispiel:

```
# zfs inherit -r mountpoint rpool/ROOT/s10up
# zfs set mountpoint=/ rpool/ROOT/s10up
```

5 Starten Sie das System neu.

Wählen Sie die BU aus, deren Einhängepunkte Sie gerade korrigiert haben, wenn im GRUB-Menü bzw. in der Eingabeaufforderung des OpenBoot-PROM die Option zum Booten einer spezifischen Boot-Umgebung angezeigt wird.

Booten zur Wiederherstellung in einer ZFS-Root-Umgebung

Gehen Sie wie folgt vor, um das System zu booten und eine Wiederherstellung durchzuführen, wenn ein `root`-Passwort verloren gegangen oder ein ähnliches Problem aufgetreten ist.

Je nach Schweregrad des Fehlers müssen Sie im Failsafe-Modus oder von einem alternativen Datenträger booten. Um ein verloren gegangenes oder unbekanntes `root`--Passwort wiederherzustellen, können Sie in der Regel im Failsafe-Modus booten.

- „So booten Sie im ZFS-Failsafe-Modus“ auf Seite 168
- „So booten Sie ZFS von einem alternativen Datenträger“ auf Seite 169

Informationen zur Wiederherstellung eines Root-Pools oder Root-Pool-Schappschusses finden Sie unter „Wiederherstellen von ZFS-Root-Pools oder Root-Pool-Snapshots“ auf Seite 169.

▼ So booten Sie im ZFS-Failsafe-Modus

1 Booten Sie im Failsafe-Modus.

- Auf einem SPARC-System geben Sie an der Eingabeaufforderung `ok` Folgendes ein:

  ```
  ok boot -F failsafe
  ```

- Auf einem x86-System wählen Sie im GRUB-Menü den Failsafe-Modus aus.

2 Hängen Sie die ZFS-BU in `/a` ein, wenn Sie dazu aufgefordert werden.

```
.
.
.
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 Wechseln Sie in das Verzeichnis `/a/etc`.

```
# cd /a/etc
```

4 Setzen Sie nötigenfalls den TERM-Typ.

```
# TERM=vt100
# export TERM
```

5 Korrigieren Sie die Datei `passwd` bzw. `shadow`.

```
# vi shadow
```

6 Starten Sie das System neu.

```
# init 6
```

▼ So booten Sie ZFS von einem alternativen Datenträger

Wenn ein Problem das erfolgreiche Booten des Systems verhindert oder ein anderes schwerwiegendes Problem auftritt, müssen Sie von einem Netzwerkinstallationsserver oder von einer Oracle Solaris-Installations-DVD booten, den Root-Pool importieren, die ZFS-BU einhängen und anschließend versuchen, das Problem zu lösen.

1. **Booten Sie von einer Installations-DVD oder über das Netzwerk.**
   - SPARC – Wählen Sie eine der folgenden Boot-Methoden:

     ```
     ok boot cdrom -s
     ok boot net -s
     ```

     Wenn Sie die Option `-s` nicht verwenden, müssen Sie das Installationsprogramm beenden.
   - x86 – Wählen Sie die Option zum Booten über das Netzwerk oder booten Sie von einer lokalen DVD.
2. **Importieren Sie den Root-Pool und geben Sie einen alternativen Einhängepunkt an. Beispiel:**

   ```
   # zpool import -R /a rpool
   ```
3. **Hängen Sie die ZFS-BU ein. Beispiel:**

   ```
   # zfs mount rpool/ROOT/zfsBE
   ```
4. **Greifen Sie über das Verzeichnis `/a` auf den ZFS-BU-Inhalt zu.**

   ```
   # cd /a
   ```
5. **Starten Sie das System neu.**

   ```
   # init 6
   ```

Wiederherstellen von ZFS-Root-Pools oder Root-Pool-Snapshots

In den folgenden Abschnitten werden diese Vorgehensweisen beschrieben:

- „So ersetzen Sie eine Festplatte im ZFS-Root-Pool“ auf Seite 170
- „So erstellen Sie Root-Pool-Schnappschüsse“ auf Seite 172
- „So erstellen Sie einen ZFS-Root-Pool neu und stellen Root-Pool-Schnappschüsse wieder her“ auf Seite 174
- „So erstellen Sie nach dem Booten im Failsafe-Modus ein Dateisystem im Zustand eines früheren Schnappschusses wieder her“ auf Seite 175

▼ So ersetzen Sie eine Festplatte im ZFS-Root-Pool

Das Ersetzen einer Festplatte im Root-Pool kann aus folgenden Gründen erforderlich sein:

- Der Root-Pool ist zu klein und Sie möchten eine kleine durch eine größere Festplatte ersetzen.
- Eine Root-Pool-Festplatte ist ausgefallen. Wenn Sie einen nicht redundanten Pool verwenden und nach einem Festplattenausfall das System nicht mehr booten können, müssen Sie von anderen Medien wie einer DVD oder dem Netzwerk booten, bevor Sie die Root-Pool-Festplatte ersetzen können.

In einer gespiegelten Root-Pool-Konfiguration können Sie versuchen, eine Festplatte zu ersetzen, ohne von einem alternativen Medium zu booten. Sie können eine ausgefallene Festplatte ersetzen, indem Sie den Befehl `zpool replace` verwenden. Wenn Sie über eine zusätzliche Festplatte verfügen, können Sie auch den Befehl `zpool attach` verwenden. In diesem Abschnitt finden Sie ein Beispiel für das Verfahren zum Einbinden einer zusätzlichen Festplatte und zum Entfernen einer Root-Pool-Festplatte.

Bei mancher Hardware müssen Sie eine Festplatte zunächst deaktivieren und dekonfigurieren, bevor Sie versuchen können, eine ausgefallene Festplatte mithilfe des Vorgangs `zpool replace` zu ersetzen. Beispiel:

```
# 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
```

Bei mancher Hardware müssen Sie die Ersatzfestplatte nach dem Einsetzen weder aktivieren noch neu konfigurieren.

Sie müssen die Pfadnamen des Boot-Geräts der aktuellen und der neuen Festplatte angeben, damit Sie das Booten von der Ersatzfestplatte testen und manuell von der vorhandenen Festplatte booten können, falls das Booten von der Ersatzfestplatte fehlschlägt. In dem Beispiel des folgenden Verfahrens lautet der Pfadname für die aktuelle Root-Pool-Festplatte (`c1t10d0s0`):

```
/pci@8,700000/pci@3/scsi@5/sd@a,0
```

Der Pfadname für die Ersatzfestplatte zum Booten (`c1t9d0s0`) lautet:

```
/pci@8,700000/pci@3/scsi@5/sd@9,0
```

1. **Verbinden Sie die Ersatzfestplatte (bzw. die neue Festplatte) physisch.**

2. **Vergewissern Sie sich, dass die neue Festplatte ein SMI-Label und den Bereich 0 hat.**

   Informationen zum Umbenennen einer für den Root-Pool bestimmten Festplatte finden Sie in den folgenden Referenzen:

   - SPARC: „How to Set Up a Disk for a ZFS Root File System“ in *System Administration Guide: Devices and File Systems*
   - x86: „How to Set Up a Disk for a ZFS Root File System“ in *System Administration Guide: Devices and File Systems*

3. **Verbinden Sie die neue Festplatte mit dem Root-Pool.**

   Beispiel:

   ```
   # zpool attach rpool c1t10d0s0 c1t9d0s0
   ```

4. **Überprüfen Sie den Status des Root-Pools.**

   Beispiel:

   ```
   # 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. **Wenn Sie eine kleinere Root-Poolfestplatte durch eine größere Festplatte ersetzen, legen Sie die Eigenschaft `autoexpand` des Pools so fest, dass die Poolgröße erweitert wird.**

   Bestimmen Sie die vorhandene `rpool`-Poolgröße:

   ```
   # zpool list rpool
   NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
   rpool  29.8G   152K  29.7G   0%  1.00x  ONLINE  -

   # zpool set autoexpand=on rpool
   ```

   Prüfen Sie die erweiterte `rpool`-Poolgröße:

   ```
   # zpool list rpool
   NAME   SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
   rpool  279G   146K  279G   0%  1.00x  ONLINE  -
   ```

6 **Überprüfen Sie, ob Sie von der neuen Festplatte booten können.**

Auf einem SPARC-System würden Sie eine Syntax wie die folgende verwenden:

```
ok boot /pci@8,700000/pci@3/scsi@5/sd@9,0
```

7 **Wenn das System von der neuen Festplatte bootet, entfernen Sie die alte Festplatte.**

Beispiel:

```
# zpool detach rpool c1t10d0s0
```

8 **Richten Sie das System so ein, dass es automatisch von der neuen Festplatte bootet, indem Sie das Standard-Boot-Gerät zurücksetzen.**

- SPARC – Verwenden Sie den Befehl `eeprom` oder `setenv` vom SPARC-Boot-PROM.
- x86 – Konfigurieren Sie das System-BIOS erneut.

▼ So erstellen Sie Root-Pool-Schnappschüsse

Sie können Root-Pool-Schnappschüsse zur Wiederherstellung erstellen. Der beste Weg zum Erstellen von Root-Pool-Schnappschüssen besteht darin, einen rekursiven Schnappschuss des Root-Pools zu erstellen.

Das folgende Verfahren erstellt einen rekursiven Root-Pool-Schnappschuss und speichert diesen sowohl als Datei als auch als Schnappschüsse in einem Pool auf einem entfernten System. Wenn ein Root-Pool ausfällt, kann das entfernte Dataset durch Verwendung von NFS eingehängt und die Schnappschussdatei im wiederhergestellten Pool gespeichert werden. Sie können Root-Pool-Schnappschüsse auch als die eigentlichen Schnappschüsse in einem Pool auf einem entfernten System speichern. Das Senden und Empfangen von Schnappschüssen über ein entferntes System ist etwas komplexer, da Sie `ssh` konfigurieren oder `rsh` verwenden müssen, während das zu reparierende System von dem Miniroot des Betriebssystems Oracle Solaris gebootet wird.

Das Validieren von Schnappschüssen, die auf entfernten Systemen als Dateien oder Schnappschüsse gespeichert sind, ist ein wichtiger Schritt bei der Wiederherstellung von Root-Pools. Bei beiden Methoden sollten Schnappschüsse regelmäßig wiederhergestellt werden, beispielsweise, wenn sich die Konfiguration des Pools verändert oder das Solaris-Betriebssystem aktualisiert wird.

Im folgenden Verfahren wird das System von der Boot-Umgebung `zfsBE` gebootet.

1 **Erstellen Sie ein Pool- und Dateisystem zum Speichern von Schnappschüssen auf einem entfernten System.**

Beispiel:

```
remote# zfs create rpool/snaps
```

2 Geben Sie das Dateisystem für das lokale System frei.

Beispiel:

```
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 Erstellen Sie einen rekursiven Snapshot des Root-Pools.

```
local# zfs snapshot -r rpool@snap1
local# zfs list -r rpool
# NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    15.1G   119G   106K  /rpool
rpool@snap1                  0      -   106K  -
rpool/ROOT               5.00G   119G    31K  legacy
rpool/ROOT@snap1             0      -    31K  -
rpool/ROOT/zfsBE         5.00G   119G  5.00G  /
rpool/ROOT/zfsBE@snap1       0      -  5.00G  -
rpool/dump               2.00G   120G  1.00G  -
rpool/dump@snap1             0      -  1.00G  -
rpool/export               63K   119G    32K  /export
rpool/export@snap1           0      -    32K  -
rpool/export/home          31K   119G    31K  /export/home
rpool/export/home@snap1      0      -    31K  -
rpool/swap               8.13G   123G  4.00G  -
rpool/swap@snap1             0      -  4.00G  -
```

4 Senden Sie die Root-Pool-Snapshots an das entfernte System.

Um beispielsweise die Root-Pool-Schnappschüsse als Datei an einen entfernten Pool zu senden, verwenden Sie folgende Syntax:

```
local# zfs send -Rv rpool@snap1 > /net/remote-system/rpool/snaps/rpool.snap1
sending from @ to rpool@snap1
sending from @ to rpool/ROOT@snap1
sending from @ to rpool/ROOT/s10zfsBE@snap1
sending from @ to rpool/dump@snap1
sending from @ to rpool/export@snap1
sending from @ to rpool/export/home@snap1
sending from @ to rpool/swap@snap1

local# zfs send -Rv rpool@snap1 > /net/remote-system/rpool/snaps/rpool.snap1
sending from @ to rpool@snap1
sending from @ to rpool/export@snap1
sending from @ to rpool/export/home@snap1
sending from @ to rpool/ROOT@snap1
sending from @ to rpool/ROOT/zfsBE@snap1
sending from @ to rpool/dump@snap1
sending from @ to rpool/swap@snap1
```

Wenn Sie die Root-Pool-Schnappschüsse als Schnappschüsse an einen entfernten Pool senden möchten, verwenden Sie folgende Syntax:

```
local# zfs send -Rv rpool@snap1 | ssh remote-system zfs receive
-Fd -o canmount=off tank/snaps
sending from @ to rpool@snap1
sending from @ to rpool/export@snap1
```

```
sending from @ to rpool/export/home@snap1
sending from @ to rpool/ROOT@snap1
sending from @ to rpool/ROOT/zfsBE@snap1
sending from @ to rpool/dump@snap1
sending from @ to rpool/swap@snap1
```

▼ So erstellen Sie einen ZFS-Root-Pool neu und stellen Root-Pool-Schnappschüsse wieder her

In diesem Verfahren wird von folgenden Voraussetzungen ausgegangen:

- Der ZFS-Root-Pool kann nicht wiederhergestellt werden.
- Die ZFS-Root-Pool-Schnappschüsse wurden auf einem entfernten System gespeichert und über NFS freigegeben.

Alle Schritte werden auf dem lokalen System ausgeführt.

1 Booten Sie von einer Installations-DVD oder über das Netzwerk.

- SPARC – Wählen Sie eine der folgenden Boot-Methoden:

  ```
  ok boot net -s
  ok boot cdrom -s
  ```

  Wenn Sie die Option `-s` nicht verwenden, müssen Sie das Installationsprogramm beenden.
- x86 – Wählen Sie die Option zum Booten von der DVD oder über das Netzwerk. Beenden Sie das Installationsprogramm.

2 Hängen Sie das entfernte Schnappschuss-Dateisystem ein, wenn Sie die Root-Pool-Schnappschüsse als Datei an das entfernte System gesendet haben.

Beispiel:

```
# mount -F nfs remote-system:/rpool/snaps /mnt
```

Wenn Ihre Netzwerkservices nicht konfiguriert sind, müssen Sie eventuell die IP-Adresse des *remote-system* angeben.

3 Wenn die Root-Pool-Festplatte ersetzt wird und keine Festplattenbezeichnung enthält, die von ZFS verwendet werden kann, müssen Sie die Festplatte umbenennen.

Weitere Informationen über die Neubezeichnung der Festplatte finden Sie in den folgenden Referenzen:

- SPARC: „How to Set Up a Disk for a ZFS Root File System“ in *System Administration Guide: Devices and File Systems*
- x86: „How to Set Up a Disk for a ZFS Root File System“ in *System Administration Guide: Devices and File Systems*

4 **Erstellen Sie den Root-Pool neu.**

Beispiel:

```
# zpool create -f -o failmode=continue -R /a -m legacy -o cachefile=
/etc/zfs/zpool.cache rpool c1t1d0s0
```

5 **Stellen Sie die Root-Pool-Schnappschüsse wieder her.**

Dieser Schritt kann etwas Zeit beanspruchen. Beispiel:

```
# cat /mnt/rpool.snap1 | zfs receive -Fdu rpool
```

Durch Verwenden der Option `-u` wird das wiederhergestellte Archiv nach dem `zfs receive`-Vorgang nicht eingehängt.

Um die tatsächlich in einem Pool auf einem entfernten System gespeicherten Root-Pool-Schnappschüsse wiederherzustellen, verwenden Sie folgende Syntax:

```
# ssh remote-system zfs send -Rb tank/snaps/rpool@snap1 | zfs receive -F rpool
```

6 **Überprüfen Sie, ob die Root-Pool-Datasets wiederhergestellt wurden.**

Beispiel:

```
# zfs list
```

7 **Legen Sie die Eigenschaft `bootfs` in der Root-Pool-BU fest.**

Beispiel:

```
# zpool set bootfs=rpool/ROOT/zfsBE rpool
```

8 **Installieren Sie auf der neuen Festplatte die Boot-Blöcke.**

- 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 **Starten Sie das System neu.**

```
# init 6
```

▼ So erstellen Sie nach dem Booten im Failsafe-Modus ein Dateisystem im Zustand eines früheren Schnappschusses wieder her

Für dieses Verfahren wird vorausgesetzt, dass Root-Pool-Schnappschüsse verfügbar sind. In diesem Beispiel befinden sie sich auf dem lokalen System.

```
# zfs snapshot -r rpool@snap1
# zfs list -r rpool
NAME                         USED  AVAIL  REFER  MOUNTPOINT
rpool                       7.84G  59.1G   109K  /rpool
rpool@snap1                   21K      -   106K  -
rpool/ROOT                  4.78G  59.1G    31K  legacy
rpool/ROOT@snap1               0      -    31K  -
rpool/ROOT/s10zfsBE         4.78G  59.1G  4.76G  /
rpool/ROOT/s10zfsBE@snap1   15.6M      -  4.75G  -
rpool/dump                  1.00G  59.1G  1.00G  -
rpool/dump@snap1              16K      -  1.00G  -
rpool/export                  99K  59.1G    32K  /export
rpool/export@snap1            18K      -    32K  -
rpool/export/home             49K  59.1G    31K  /export/home
rpool/export/home@snap1       18K      -    31K  -
rpool/swap                  2.06G  61.2G    16K  -
rpool/swap@snap1               0      -    16K  -
```

1 Fahren Sie das System herunter und booten Sie im Failsafe-Modus.

```
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 Stellen Sie jeden Root-Pool-Schnappschuss wieder her.

```
# zfs rollback rpool@snap1
# zfs rollback rpool/ROOT@snap1
# zfs rollback rpool/ROOT/s10zfsBE@snap1
```

3 Booten Sie erneut im Mehrbenutzer-Modus.

```
# init 6
```

◆ ◆ ◆ KAPITEL 5

Verwalten von Oracle Solaris ZFS-Dateisystemen

Dieses Kapitel enthält ausführliche Informationen zum Verwalten von Oracle Solaris ZFS-Dateisystemen. Es werden Konzepte wie die hierarchische Dateisystemstrukturierung, Eigenschaftsvererbung, die automatische Administration von Einhängepunkten sowie die Interaktion zwischen Netzwerkdateisystemen behandelt.

Dieses Kapitel enthält die folgenden Abschnitte:

Verwalten von ZFS-Dateisystemen (Übersicht)

ZFS-Dateisysteme setzen auf einem Speicher-Pool auf. Dateisysteme können dynamisch erstellt und gelöscht werden, ohne dass Festplattenspeicher zugewiesen oder formatiert werden muss. Da Dateisysteme so kompakt und der Dreh- und Angelpunkt der ZFS-Administration sind, werden Sie wahrscheinliche sehr viele davon erstellen.

ZFS-Dateisysteme werden mit dem Befehl `zfs` verwaltet. Der Befehl `zfs` enthält eine Reihe von Unterbefehlen, die spezifische Operationen an Dateisystemen ausführen. In diesem Kapitel werden diese Unterbefehle ausführlich beschrieben. Schnappschüsse, Volumes und Klone werden von diesem Befehl ebenfalls verwaltet; die Leistungsmerkmale werden jedoch erst später behandelt. Ausführliche Informationen zu Schnappschüssen und Klonen finden Sie in Kapitel 6, „Arbeiten mit Oracle Solaris ZFS-Schnappschüssen und -Klonen“. Ausführliche Informationen zu ZFS-Volumes finden Sie unter „ZFS-Volumes“ auf Seite 283.

Hinweis – Der Begriff *Dataset* wird in diesem Kapitel als Oberbegriff für Dateisysteme, Snapshots, Klone und Volumes verwendet.

Erstellen, Entfernen und Umbenennen von ZFS-Dateisystemen

ZFS-Dateisysteme können mit den Befehlen `zfs create` bzw. `zfs destroy` erstellt und gelöscht werden. ZFS-Dateisysteme können mit dem Befehl `zfs rename` umbenannt werden.

Erstellen eines ZFS-Dateisystems

ZFS-Dateisysteme werden mit dem Befehl `zfs create` erstellt. Der Befehl `create` erfordert als einziges Argument den Namen des zu erstellenden Dateisystems. Der Name des Dateisystems wird wie folgt als Pfadname beginnend mit dem Namen des Pools angegeben:

Pool-Name/[Dateisystemname/]Dateisystemname

Der Pool-Name und die anfänglichen Dateisystemnamen im Pfad geben an, wo das neue Dateisystem in der Hierarchie erstellt wird. Der letzte Name im Pfad ist der Name des zu erstellenden Dateisystems. Der Dateisystemname muss den unter „Konventionen für das Benennen von ZFS-Komponenten“ auf Seite 31 aufgeführten Benennungskonventionen entsprechen.

Im folgenden Beispiel wird das Dateisystem `jeff` im Dateisystem `tank/home` erstellt.

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

ZFS hängt das neue Dateisystem bei fehlerfreier Erstellung automatisch ein. Dateisysteme werden standardmäßig als */Dataset* eingehängt; dabei wird der im Unterbefehl `create` angegebene Pfad verwendet. In diesem Beispiel wird das neu erstellte Dateisystem `jeff` unter `/tank/home/jeff` eingehängt. Weitere Informationen zu automatisch verwalteten Einhängepunkten finden Sie unter „Verwalten von ZFS-Einhängepunkten“ auf Seite 204.

Weitere Informationen zum Befehl `zfs create` finden Sie in der Man Page `zfs(1M)`.

Sie können Dateisystemeigenschaften beim Erstellen des Dateisystems festlegen.

Im folgenden Beispiel wird ein Einhängepunkt von `/export/zfs` für das Dateisystem `tank/home` erstellt:

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

Weitere Informationen zu Eigenschaften von Dateisystemen finden Sie unter „ZFS-Eigenschaften“ auf Seite 181.

Löschen eines ZFS-Dateisystems

ZFS-Dateisysteme werden mit dem Befehl `zfs destroy` gelöscht. Das gelöschte Dateisystem wird automatisch für den Netzwerkzugriff gesperrt und ausgehängt. Weitere Informationen zur automatischen Administration von Einhängepunkten und gemeinsam genutzten Objekten finden Sie unter „Automatische Einhängepunkte“ auf Seite 204.

Im folgenden Beispiel wird das Dateisystem `tank/home/mark` gelöscht:

```
# zfs destroy tank/home/mark
```

Achtung – Beim Ausführen des Unterbefehls `destroy` wird keine Bestätigung des Löschvorgangs angefordert. Verwenden Sie diesen Befehl deshalb mit äußerster Vorsicht.

Wenn das zu löschende Dateisystem noch von Ressourcen verwendet wird und deswegen nicht ausgehängt werden kann, schlägt der Befehl `zfs destroy` fehl. Aktive Dateisysteme werden mit der Option `-f` gelöscht. Sie sollten diese Option mit Sorgfalt verwenden, da sie aktive Dateisysteme aushängt, für den Netzwerkzugriff sperrt und löscht und somit unvorgesehenes Systemverhalten verursachen kann.

```
# zfs destroy tank/home/matt
cannot unmount 'tank/home/matt': Device busy

# zfs destroy -f tank/home/matt
```

Der Befehl `zfs destroy` schlägt ebenfalls fehl, wenn in einem Dateisystem untergeordnete Dateisysteme vorhanden sind. Zum rekursiven Löschen von Dateisystemen und allen untergeordneten Dateisystemen dient die Option `-r`. Bitte beachten Sie, dass beim rekursiven Löschen auch Schnappschüsse des Dateisystems gelöscht werden. Deshalb sollten Sie diese Option mit äußerster Vorsicht verwenden.

```
# zfs destroy tank/ws
cannot destroy 'tank/ws': filesystem has children
use '-r' to destroy the following datasets:
tank/ws/jeff
tank/ws/bill
tank/ws/mark
# zfs destroy -r tank/ws
```

Wenn das zu löschende Dateisystem indirekte untergeordnete Dateisysteme besitzt, schlägt auch der rekursive Löschbefehl fehl. Wenn Sie das Löschen *aller* untergeordneten Objekte einschließlich geklonter Dateisysteme außerhalb der Zielhierarchie erzwingen wollen, müssen Sie die Option `-R` verwenden. Verwenden Sie diese Option mit äußerster Vorsicht.

```
# zfs destroy -r tank/home/eric
cannot destroy 'tank/home/eric': filesystem has dependent clones
use '-R' to destroy the following datasets:
tank//home/eric-clone
# zfs destroy -R tank/home/eric
```

Achtung – Für die Optionen `-f`, `-r` und `-R` des Löschbefehls `zfs destroy` wird keine Bestätigung angefordert. Deshalb sollten Sie diese Optionen mit äußerster Vorsicht verwenden.

Weitere Informationen zu Schnappschüssen und Klonen finden Sie in Kapitel 6, „Arbeiten mit Oracle Solaris ZFS-Schnappschüssen und -Klonen“.

Umbenennen eines ZFS-Dateisystems

Dateisysteme können mit dem Befehl `zfs rename` umbenannt werden. Mit dem Unterbefehl `rename` können folgende Vorgänge ausgeführt werden:

- Ändern des Namens eines Dateisystems.
- Verlagern eines Dateisystems innerhalb der ZFS-Hierarchie
- Ändern des Namens eines Dateisystems und Verlagern dieses Systems innerhalb der ZFS-Hierarchie

Im folgenden Beispiel wird der Unterbefehl `rename` verwendet, um ein Dateisystem von `eric` in `eric_old` umzubenennen:

```
# zfs rename tank/home/eric tank/home/eric_old
```

Das folgende Beispiel zeigt die Verwendung des Befehls `zfs rename` zum Verlagern eines ZFS-Dateisystems:

```
# zfs rename tank/home/mark tank/ws/mark
```

In diesem Beispiel wird das Dateisystem `mark` von `tank/home` nach `tank/ws` verlagert. Wenn ein Dateisystem mithilfe des Umbenennungsbefehls verlagert wird, muss sich der neue Speicherort innerhalb des gleichen Pools befinden, und dieser muss über genügend Festplattenkapazität für das Dateisystem verfügen. Wenn der neue Speicherort nicht genügend Festplattenkapazität besitzt (z.B. weil das zugeteilte Kontingent erreicht ist), verläuft der `rename`-Vorgang nicht erfolgreich.

Weitere Informationen zu Kontingenten finden Sie unter „Einstellen von ZFS-Kontingenten und -Reservierungen“ auf Seite 210.

Durch die Umbenennung wird das betreffende Dateisystem mit allen seinen untergeordneten Dateisystemen ausgehängt und wieder neu eingehängt. Die Umbenennung schlägt fehl, wenn ein aktives Dateisystem nicht ausgehängt werden kann. Wenn dieses Problem auftritt, müssen Sie das Aushängen des Dateisystems erzwingen.

Informationen zum Umbenennen von Snapshots finden Sie unter „Umbenennen von ZFS-Schnappschüssen“ auf Seite 221.

ZFS-Eigenschaften

Mithilfe von Eigenschaften kann das Verhalten von Dateisystemen, Volumes, Schnappschüssen und Klonen gesteuert werden Wenn keine anderen Angaben gemacht werden, gelten die in diesem Abschnitt definierten Eigenschaften für alle Dataset-Typen.

Eigenschaften werden in zwei Typen (native und benutzerdefinierte Eigenschaften) eingeteilt. Native Eigenschaften stellen interne Statistikinformationen bereit oder kontrollieren das Systemverhalten von ZFS-Dateisystemen. Darüber hinaus können native Eigenschaften konfigurierbar oder schreibgeschützt sein. Benutzerdefinierte Eigenschaften wirken sich nicht auf das Verhalten von ZFS-Dateisystemen aus, können jedoch zum Versehen von Datasets mit Informationen, die für Ihre lokalen Gegebenheiten wichtig sind, verwendet werden. Weitere Informationen zu benutzerdefinierten Eigenschaften finden Sie unter „Benutzerdefinierte ZFS-Eigenschaften“ auf Seite 194.

Die meisten konfigurierbaren Eigenschaften sind vererbbar. Vererbbare Eigenschaften werden von einem übergeordneten Dateisystem an alle seine untergeordneten Dateisysteme weitergegeben.

Alle vererbbaren Eigenschaften haben eine Quelle, die angibt, wie eine Eigenschaft erhalten wurde. Die Eigenschaftsquelle kann die folgenden Werte besitzen:

`local`	Dieser Wert zeigt an, dass die Eigenschaft mithilfe des Befehls `zfs set` (siehe „Setzen von ZFS-Eigenschaften“ auf Seite 198) explizit für das Dataset festgelegt wurde.
`inherited from` *Dataset-Name*	Dieser Wert zeigt an, dass die Eigenschaft von einem übergeordneten Objekt geerbt wurde.
`default`	Dieser Wert zeigt an, dass der Wert der betreffenden Eigenschaft weder geerbt noch lokal gesetzt wurde. Diese Quelle resultiert daraus, dass kein übergeordnetes Objekt eine Eigenschaft aufweist, die mit source `local` definiert ist.

In der folgenden Tabelle sind schreibgeschützte und konfigurierbare native Eigenschaften von ZFS-Dateisystemen aufgeführt. Schreibgeschützte Eigenschaften sind entsprechend gekennzeichnet. Alle anderen in dieser Tabelle aufgeführten nativen Eigenschaften sind konfigurierbar. Weitere Informationen zu benutzerdefinierten Eigenschaften finden Sie unter „Benutzerdefinierte ZFS-Eigenschaften“ auf Seite 194.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften

Eigenschaft	Typ	Standardwert	Beschreibung
`aclinherit`	Zeichenkette	`secure`	Legt fest, wie Einträge in Zugriffskontrolllisten beim Erstellen von Dateien und Verzeichnissen vererbt werden. Werte: `discard`, `noallow`, `secure` und `passthrough`. Eine Beschreibung dieser Werte finden Sie in „Eigenschaften von Zugriffskontrolllisten“ auf Seite 248.
`aclmode`	Zeichenkette	`groupmask`	Kontrolliert wie ein Zugriffskontrolllisteneintrag während eines `chmod`-Vorgangs geändert wird. Werte: `discard`, `groupmask` und `passthrough`. Eine Beschreibung dieser Werte finden Sie in „Eigenschaften von Zugriffskontrolllisten“ auf Seite 248.
`atime`	Boolesch	`on`	Legt fest, ob beim Lesen von Dateien die Dateizugriffszeit aktualisiert wird. Durch Deaktivierung dieser Eigenschaft wird vermieden, dass während des Lesens von Dateien Datenverkehr entsteht, der aus Schreibvorgängen resultiert. Dadurch kann die Leistung erheblich verbessert werden. E-Mail-Programme und ähnliche Serviceprogramme können allerdings in ihrer Funktion beeinträchtigt werden.
`available`	Zahl	entf.	Diese schreibgeschützte Eigenschaft gibt die für ein Dateisystem und alle seine untergeordneten Objekte verfügbare Festplattenkapazität an. Dabei wird angenommen, dass im Pool keine Aktivität vorliegt. Da Festplattenkapazität innerhalb eines Pools gemeinsam genutzt wird, kann die verfügbare Kapazität durch verschiedene Faktoren wie z. B. physische Speicherkapazität des Pools, Kontingente, Reservierungen oder andere im Pool befindliche Datasets beschränkt werden. Die Abkürzung der Eigenschaft lautet `avail`. Weitere Informationen zur Berechnung von Festplattenkapazität finden Sie unter „Berechnung von ZFS-Festplattenkapazität“ auf Seite 32.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`canmount`	Boolesch	`on`	Legt fest, ob ein Dateisystem mit dem Befehl `zfs mount` eingehängt werden kann. Diese Eigenschaft ist für jedes Dateisystem einstellbar und kann nicht vererbt werden. Durch Setzen dieser Eigenschaft auf `off` kann jedoch an untergeordnete Dateisysteme ein Einhängepunkt vererbt werden. Die Dateisysteme selbst werden jedoch nicht eingehängt. Weitere Informationen dazu finden Sie unter „Die Eigenschaft `canmount`“ auf Seite 193.
`checksum`	Zeichenkette	`on`	Aktiviert/deaktiviert die Prüfsumme zur Validierung der Datenintegrität. Der Standardwert ist `on`. Dadurch wird automatisch ein geeigneter Algorithmus (gegenwärtig `fletcher4`) gesetzt. Werte: `on`, `off`, `fletcher2` , `fletcher4` und `sha256`. Der Wert `off` deaktiviert die Integritätsprüfung von Benutzerdaten. Ein Wert `off` wird nicht empfohlen.
`compression`	Zeichenkette	`off`	Aktiviert oder deaktiviert die Komprimierung für ein Dataset. Werte: `on`, `off`, `lzjb`, `gzip` und `gzip-` *N*. Derzeit hat das Setzen der Eigenschaft auf `lzjb`, `gzip` oder `gzip-` *N* dieselbe Wirkung wie die Einstellung auf `on`. Durch Aktivieren der Komprimierung an einem Dateisystem mit bereits vorhandenen Daten werden nur neu hinzugekommene Daten komprimiert. Vorhandene Daten bleiben unkomprimiert. Die Abkürzung der Eigenschaft lautet `compress`
`compressratio`	Zahl	entf.	Schreibgeschützte Eigenschaft, die das für das betreffende Dataset erreichte Komprimierungsverhältnis als Faktor ausdrückt. Die Komprimierung kann mit dem Befehl `zfs set compression=on` *dataset* aktiviert werden. Der Wert wird aus der logischen Kapazität aller Dateien und der Kapazität der entsprechend referenzierten physischen Daten berechnet. Explizite Speicherplatzeinsparungen durch die Eigenschaft `compression` sind in diesem Wert enthalten.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`copies`	Zahl	`1`	Legt die Anzahl der Kopien von Benutzerdaten pro Dateisystem fest. Verfügbare Werte sind `1`, `2` oder `3`. Diese Kopien werden zusätzlich zu etwaigen Redundanzfunktionen auf Pool-Ebene angelegt. Die von zusätzlichen Kopien beanspruchte Festplattenkapazität wird auf die entsprechende Datei bzw. das Dataset angerechnet und zählt für Kontingente und Reservierungen. Darüber hinaus wird die Eigenschaft `used` entsprechend aktualisiert, wenn das Erstellen mehrerer Kopien aktiviert wurde. Sie sollten diese Eigenschaft beim Erstellen des Dateisystems setzen, da sich das Ändern dieser Eigenschaft bei einem bereits vorhandenen Dateisystem nur auf neu geschriebene Daten auswirkt.
`creation`	Zeichenkette	entf.	Schreibgeschützte Eigenschaft, die angibt, wann ein Dataset erstellt wurde (Datum/Uhrzeit).
`devices`	Boolesch	`on`	Bestimmt, ob die Gerätedateien in einem Dateisystem geöffnet werden können.
`exec`	Boolesch	`on`	Legt fest, ob Programme innerhalb eines Dateisystems ausführbar sind. Wenn diese Eigenschaft auf `off` gesetzt ist, werden `mmap(2)`-Aufrufe mit `PROT_EXEC` nicht zugelassen.
`mounted`	Boolesch	entf.	Schreibgeschützte Eigenschaft, die angibt, ob gegenwärtig ein Dateisystem, Klon oder Schnappschuss eingehängt ist. Diese Eigenschaft gilt nicht für Volumes. Werte: `yes` oder `no`.
`mountpoint`	Zeichenkette	entf.	Legt den Einhängepunkt für das betreffende Dateisystem fest. Wenn die Eigenschaft `mountpoint` für ein Dateisystem geändert wird, werden das Dateisystem selbst und alle seine untergeordneten Dateisysteme, die den Einhängepunkt geerbt haben, ausgehängt. Wenn der neue Wert `legacy` ist, bleiben sie ausgehängt. Andernfalls werden sie, wenn die Eigenschaft vorher auf `legacy` oder `none` gesetzt war bzw. die Dateisysteme vor dem Ändern der Eigenschaft eingehängt waren, am neuen Bestimmungsort automatisch eingehängt. Darüber hinaus wird der Netzwerkzugriff auf die betreffenden Dateisysteme gesperrt und am neuen Bestimmungsort freigegeben. Weitere Informationen zur Verwendung dieser Eigenschaft finden Sie unter „Verwalten von ZFS-Einhängepunkten“ auf Seite 204.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`primarycache`	Zeichenkette	`all`	Kontrolliert, was im Primär-Cache gespeichert wird (ARC). Mögliche Werte sind `all`, `none` und `metadata`. Ist diese Eigenschaft auf `all` gesetzt, werden sowohl Benutzerdaten als auch Metadaten im Cache gespeichert. Ist diese Eigenschaft auf `none` gesetzt, werden weder Benutzerdaten noch Metadaten im Cache gespeichert. Ist diese Eigenschaft auf `metadata` gesetzt, werden nur Metadaten im Cache gespeichert. Sobald diese Eigenschaften für bereits vorhandene Dateisysteme eingestellt werden, werden nur neue E/A-Vorgänge auf der Basis des Werts dieser Eigenschaften im Cache gespeichert. Bei einigen Datenbank-Umgebungen kann es von Vorteil sein, Benutzerdaten nicht im Cache zu speichern. Sie müssen ermitteln, ob die Festlegung von Cacheeigenschaften für Ihre Umgebung zutreffend ist.
`origin`	Zeichenkette	entf.	Schreibgeschützte Eigenschaft für geklonte Dateisysteme bzw. Volumes, die angibt, aus welchem Schnappschuss der betreffende Klon erstellt wurde. Das ursprüngliche Dateisystem kann (auch mit den Optionen `-r` oder `-f`) nicht gelöscht werden, solange ein Klon vorhanden ist. Nicht-geklonte Dateisysteme haben einen Ursprung von `none`.
`quota`	Zahl (oder `none`)	`none`	Beschränkt die Festplattenkapazität, die von einem Dateisystem und untergeordneten Objekten belegt werden kann. Diese Eigenschaft erzwingt einen absoluten Grenzwert der Festplattenkapazität, die belegt werden kann. Dazu zählt auch der Speicherplatz, der von untergeordneten Objekten wie Dateisystemen und Schnappschüssen belegt wird. Das Setzen eines Kontingentes für ein untergeordnetes Objekt eines Dateisystems, für das bereits ein Kontingent definiert wurde, überschreibt den vom übergeordneten Dateisystem geerbten Wert nicht, sondern setzt darüber hinaus einen zusätzlichen Grenzwert. Kontingente können nicht für Volumes eingestellt werden, da deren Eigenschaft `volsize` bereits ein Kontingent darstellt. Weitere Informationen zum Einstellen von Kontingenten finden Sie unter „Setzen von Kontingenten für ZFS-Dateisysteme“ auf Seite 211.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`readonly`	Boolesch	`off`	Legt fest, ob ein Dataset geändert werden kann. Wenn diese Eigenschaft auf `on` gesetzt ist, können keine Änderungen vorgenommen werden. Die Abkürzung der Eigenschaft lautet `rdonly`.
`recordsize`	Zahl	`128K`	Gibt eine empfohlene Blockgröße für Dateien in einem Dateisystem an. Die Abkürzung der Eigenschaft lautet `recsize`. Eine ausführliche Beschreibung finden Sie unter „Die Eigenschaft `recordsize`“ auf Seite 193.
`referenced`	Zahl	entf.	Schreibgeschützte Eigenschaft, die die Datenmenge angibt, auf die ein Dataset zugreifen kann und die mit anderen Datasets im Pool gemeinsam verwendet werden kann. Bei der Erstellung eines Schnappschusses bzw. Klons wird anfänglich die gleiche Festplattenkapazität referenziert, die der Kapazität des Dateisystems bzw. Schnappschusses entspricht, aus dem er erstellt wurde, da der Inhalt identisch ist. Die Abkürzung der Eigenschaft lautet `refer`.
`refquota`	Zahl (oder "none")	`none`	Legt fest, wie viel Festplattenkapazität ein Dataset belegen kann. Die Eigenschaft erzwingt einen absoluten Grenzwert des belegbaren Speicherplatzes. Dieser absolute Grenzwert berücksichtigt jedoch nicht die Festplattenkapazität, die von untergeordneten Objekten wie z. B. Schnappschüssen oder Klonen belegt wird.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`refreservation`	Zahl (oder "none")	`none`	Legt die garantierte Mindestfestplattenkapazität für ein Dataset fest (ohne untergeordnete Objekte, wie etwa Schnappschüsse oder Klone). Liegt die belegte Festplattenkapazität unter dem hier angegebenen Wert, wird das Dataset behandelt, als würde es den mit `refreservation` angegebenen Speicherplatz belegen. Reservierungen durch `refreservation` werden in die Berechnung der Festplattenkapazität für das diesem Dataset übergeordnete Dataset einbezogen und auf die Kontingente und Reservierung für das übergeordnete Dataset angerechnet. Wenn `refreservation` gesetzt ist, wird ein Schnappschuss nur zugelassen, wenn außerhalb dieser Reservierung genügend freier Speicherplatz im Pool vorhanden ist, um die Menge der aktuell *referenzierten* Byte im Dataset aufzunehmen. Die Abkürzung der Eigenschaft lautet `refreserv`.
`reservation`	Zahl (oder "none")	`none`	Legt die Mindestfestplattenkapazität für ein Dateisystem und seine untergeordneten Objekte fest. Liegt die belegte Festplattenkapazität unter dem hier angegebenen Wert, wird das Dateisystem behandelt, als würde es den in dieser Reservierung angegebenen Speicherplatz belegen. Reservierungen werden für die belegte Plattenkapazität des übergeordneten Dateisystems berücksichtigt, Quota und Reservierungen des übergeordneten Dateisystems werden angerechnet. Die Abkürzung der Eigenschaft lautet `reserv`. Weitere Informationen dazu finden Sie unter „Setzen von Reservierungen für ZFS-Dateisysteme“ auf Seite 214.
`secondarycache`	Zeichenkette	`all`	Kontrolliert, was im Sekundär-Cache gespeichert wird (L2ARC). Mögliche Werte sind `all`, `none` und `metadata`. Ist diese Eigenschaft auf `all` gesetzt, werden sowohl Benutzerdaten als auch Metadaten im Cache gespeichert. Ist diese Eigenschaft auf `none` gesetzt, werden weder Benutzerdaten noch Metadaten im Cache gespeichert. Ist diese Eigenschaft auf `metadata` gesetzt, werden nur Metadaten im Cache gespeichert.
`setuid`	Boolesch	`on`	Legt fest, ob das `setuid`-Bit in einem Dateisystem berücksichtigt wird.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`shareiscsi`	Zeichenkette	off	Legt fest, ob ein ZFS-Volume gemeinsam als iSCSI-Ziel genutzt wird. Eigenschaftswerte: `on`, `off` und `type=disk`. Sie können `shareiscsi=on` für ein Dateisystem setzen, sodass alle ZFS-Volumes innerhalb des Dateisystems standardmäßig freigegeben sind. Das Setzen dieser Eigenschaft für ein Dateisystem hat jedoch keine direkte Auswirkung.
`sharenfs`	Zeichenkette	`off`	Legt fest, ob das Dateisystem über NFS zugänglich ist und welche Optionen verwendet werden. Wenn diese Eigenschaft auf `on` gesetzt ist, wird der Befehl `zfs share` ohne Optionen aufgerufen. Andernfalls wird der Befehl `zfs share` mit den Optionen aufgerufen, die dem Inhalt dieser Eigenschaft entsprechen. Wenn diese Eigenschaft auf `off` gesetzt ist, wird das Dateisystem über die älteren Befehle `share` und `unshare` sowie die Datei `dfstab` verwaltet. Weitere Informationen zur Nutzung von ZFS-Dateisystemen für den Netzwerkzugang finden Sie unter „Freigeben und Sperren von ZFS-Dateisystemen“ auf Seite 208.
`snapdir`	Zeichenkette	`hidden`	Legt fest, ob das Verzeichnis `.zfs` in der Root des Dateisystems verborgen oder sichtbar ist. Weitere Informationen über die Verwendung von Schnappschüssen finden Sie in „Überblick über ZFS-Schnappschüsse“ auf Seite 217.
`type`	Zeichenkette	entf.	Schreibgeschützte Eigenschaft, die den Typ des betreffenden Datasets (`filesystem` (Dateisystem bzw. Klon), `volume` oder `snapshot`) angibt.
`used`	Zahl	entf.	Schreibgeschützte Eigenschaft, die die für ein Dataset und alle seine untergeordneten Objekte belegte Festplattenkapazität angibt. Eine ausführliche Beschreibung finden Sie unter „Die Eigenschaft `used`“ auf Seite 191.
`usedbychildren`	Zahl	`off`	Schreibgeschützte Eigenschaft, die die Festplattenkapazität angibt, die von untergeordneten Objekten dieses Datasets beansprucht wird und die beim Löschen dieser untergeordneten Objekte frei werden würde. Die Abkürzung für die Eigenschaft lautet `usedchild`

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`usedbydataset`	Zahl	`off`	Schreibgeschützte Eigenschaft, die die Festplattenkapazität angibt, die vom Dataset selbst beansprucht wird und die beim Löschen des Datasets und vorherigem Löschen aller Schnappschüsse und Entfernen von `refreservation` frei werden würde. Die Abkürzung für diese Eigenschaft ist `usedds`.
`usedbyrefreservation`	Zahl	`off`	Schreibgeschützte Eigenschaft, die die von einem `refreservation`-Set auf diesem Dataset beanspruchte Festplattenkapazität angibt, die beim Entfernen von `refreservation` frei werden würde. Die Abkürzung für die Eigenschaft ist `usedds`.
`usedbysnapshots`	Zahl	`off`	Schreibgeschützte Eigenschaft, die die von Schnappschüssen dieses Datasets beanspruchte Festplattenkapazität angibt. Insbesondere geht es dabei um die Festplattenkapazität, die beim Löschen aller Schnappschüsse des Datasets frei werden würde. Beachten Sie, dass es sich bei diesem Wert nicht einfach um die Summe der `used`-Eigenschaften der Schnappschüsse handelt, da Speicherplatz von mehreren Schnappschüssen gemeinsam genutzt werden kann. Die Abkürzung für diese Eigenschaft ist `usedsnap`.
`version`	Zahl	entf.	Die aktuelle, von der Pool-Version unabhängige Version eines Dateisystems auf der Festplatte. Diese Eigenschaft kann nur bei einer nachfolgenden Version der unterstützten Software gesetzt werden. Weitere Informationen finden Sie unter `zfs upgrade`-Befehl.
`volsize`	Zahl	entf.	Legt die logische Größe eines Volumes fest (gilt nur für Volumes). Eine ausführliche Beschreibung finden Sie unter „Die Eigenschaft `volsize`“ auf Seite 194.
`volblocksize`	Zahl	`8 kB`	Legt die Blockgröße eines Volumes fest (gilt nur für Volumes). Nach dem Schreiben von Daten auf das betreffende Volume kann die Blockgröße nicht mehr geändert werden. Deswegen muss diese Eigenschaft bei der Erstellung des Volumes gesetzt werden. Die Standardblockgröße für Volumes ist 8 KB. Es sind alle Werte zur Potenz von 2 im Bereich von 512 Byte bis 128 KB zulässig. Die Abkürzung der Eigenschaft lautet `volblock`.

TABELLE 5–1 Beschreibungen nativer ZFS-Eigenschaften *(Fortsetzung)*

Eigenschaft	Typ	Standardwert	Beschreibung
`zoned`	Boolesch	entf.	Gibt an, ob ein Dateisystem zu einer nicht globalen Zone hinzugefügt wurde. Wenn diese Eigenschaft gesetzt ist, befindet sich der Einhängepunkt nicht in der globalen Zone, und ZFS kann ein solches Dateisystem auf Anforderung nicht einhängen. Bei der Erstinstallation einer Zone wird diese Eigenschaft für alle hinzugefügten Dateisystem festgelegt. Weitere Informationen zur Verwendung von ZFS mit installierten Zonen finden Sie in „Verwendung von ZFS in einem Solaris-System mit installierten Zonen“ auf Seite 286.
`xattr`	Boolesch	`on`	Legt fest, ob für das betreffende Dateisystem erweiterte Attribute aktiviert (`on`) oder deaktiviert sind (`off`).

Schreibgeschützte native ZFS-Eigenschaften

Schreibgeschützte native Eigenschaften können gelesen, aber nicht gesetzt werden. Schreibgeschützte native Eigenschaften werden nicht vererbt. Einige native Eigenschaften gelten nur für bestimmte Dataset-Typen. In solchen Fällen ist der entsprechende Dataset-Typ in der Beschreibung in Tabelle 5–1 aufgeführt.

Die schreibgeschützten nativen Eigenschaften sind hier aufgeführt und in Tabelle 5–1 beschrieben.

- `available`
- `compressratio`
- `creation`
- `mounted`
- `origin`
- `referenced`
- `type`
- `used`

  Ausführliche Informationen finden Sie unter „Die Eigenschaft `used`“ auf Seite 191.
- `usedbychildren`
- `usedbydataset`
- `usedbyrefreservation`
- `usedbysnapshots`

Weitere Informationen zur Berechnung von Festplattenkapazität (einschließlich Informationen zu den Eigenschaften `used`, `referenced` und `available`) finden Sie unter „Berechnung von ZFS-Festplattenkapazität“ auf Seite 32.

Die Eigenschaft `used`

Die Eigenschaft `used` ist eine schreibgeschützte Eigenschaft, die die für ein Dataset und alle seine untergeordneten Objekte belegte Festplattenkapazität angibt. Dieser Wert wird gegen die für das Dataset gesetzten Kontingente und Reservierungen geprüft. Die belegte Festplattenkapazität enthält nicht die Reservierungen für das Dataset selbst, schließt jedoch die Reservierungen für untergeordnete Datasets ein. Die Festplattenkapazität, die das Dataset aufgrund seines übergeordneten Datasets belegt, sowie die Festplattenkapazität, die beim rekursiven Löschen des Datasets freigegeben wird, sind größer als die Kapazität, die das Dataset für den belegten Speicherplatz und die Reservierung benötigt.

Bei der Erstellung von Schnappschüssen wird diese Festplattenkapazität anfänglich vom Schnappschuss und dem Dateisystem (sowie eventuellen früheren Schnappschüssen) gemeinsam genutzt. Wenn sich ein Dateisystem mit der Zeit ändert, gehört zuvor gemeinsam genutzte Festplattenkapazität dann ausschließlich dem Schnappschuss und wird in die Berechnung des vom Schnappschuss belegten Speicherplatzes einbezogen. Wie viel Festplattenkapazität von einem Schnappschuss belegt wird, hängt von den speziellen Daten des Schnappschusses ab. Zudem kann durch das Löschen von Schnappschüssen die Festplattenkapazität, die Schnappschüssen eindeutig zugewiesen ist (und von diesen verwendet wird), größer werden. Weitere Informationen zu Schnappschüssen und Speicherplatzaspekten finden Sie in „Verhalten bei ungenügendem Speicherplatz“ auf Seite 34.

In die belegte, verfügbare und referenzierte Festplattenkapazität werden keine anstehenden Änderungen einbezogen. Anstehende Änderungen werden im Allgemeinen innerhalb weniger Sekunden abgeschlossen. Nach dem Festschreiben einer Änderung auf der Festplatte durch die Funktion `fsync(3c)` oder `O_SYNC` werden die Informationen zur belegten Festplattenkapazität nicht unbedingt sofort aktualisiert.

Die Informationen der Eigenschaften `usedbychildren`, `usedbydataset` , `usedbyrefreservation` und `usedbysnapshots` kann mit dem Befehl `zfs list -o space` angezeigt werden. Diese Eigenschaften bestimmen die Eigenschaft `used` für Festplattenkapazität, die von untergeordneten Objekten belegt wird. Weitere Informationen finden Sie in Tabelle 5–1.

Konfigurierbare native ZFS-Eigenschaften

Konfigurierbare native Eigenschaften können gelesen und gesetzt werden. Konfigurierbare native Eigenschaften werden mithilfe des Befehls `zfs set`(siehe „Setzen von ZFS-Eigenschaften“ auf Seite 198) bzw. `zfs create` (siehe „Erstellen eines ZFS-Dateisystems“ auf Seite 178) gesetzt. Außer Kontingenten und Reservierungen werden konfigurierbare

Eigenschaften vererbt. Weitere Informationen zu Kontingenten und Reservierungen finden Sie unter „Einstellen von ZFS-Kontingenten und -Reservierungen“ auf Seite 210.

Einige konfigurierbare native Eigenschaften gelten nur für bestimmte Dataset-Typen. In solchen Fällen ist der entsprechende Dataset-Typ in der Beschreibung in Tabelle 5–1 aufgeführt. Sofern nichts Anderes vermerkt ist, gilt eine Eigenschaft für alle Dataset-Typen: Dateisysteme, Volumes, Klone und Schnappschüsse.

Die konfigurierbaren Eigenschaften sind hier aufgeführt und in Tabelle 5–1 beschrieben.

- `aclinherit`

  Eine ausführliche Beschreibung finden Sie unter „Eigenschaften von Zugriffskontrolllisten“ auf Seite 248.
- `aclmode`

  Eine ausführliche Beschreibung finden Sie unter „Eigenschaften von Zugriffskontrolllisten“ auf Seite 248.
- `atime`
- `canmount`
- `checksum`
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`

  Eine ausführliche Beschreibung finden Sie unter „Die Eigenschaft `recordsize`“ auf Seite 193.
- `refquota`
- `refreservation`
- `reservation`
- `secondarycache`
- `shareiscsi`
- `setuid`
- `snapdir`

- `version`
- `volsize`

  Eine ausführliche Beschreibung finden Sie unter „Die Eigenschaft `volsize`“ auf Seite 194.
- `volblocksize`
- `zoned`
- `xattr`

Die Eigenschaft `canmount`

Wenn die Eigenschaft `canmount` auf `off` gesetzt wird, kann das Dateisystem nicht mithilfe der Befehle `zfs mount` bzw. `zfs mount -a` eingehängt werden. Das Setzen dieser Eigenschaft auf `off` gleicht dem Setzen der Eigenschaft `mountpoint` auf `none`. Der Unterschied besteht darin, dass das Dateisystem dennoch noch die normale Eigenschaft `mountpoint` besitzt, die vererbt werden kann. Sie können diese Eigenschaft beispielsweise auf `off` setzen und Eigenschaften setzen, die an untergeordnete Dateisysteme vererbt werden. Das übergeordnete Dateisystem selbst wird jedoch nicht eingehängt und ist nicht für Benutzer zugänglich. In diesem Fall dient das übergeordnete Dateisystem als *Container*, für den Sie Eigenschaften festlegen können. Der Container selbst ist jedoch nicht zugänglich.

Im folgenden Beispiel wird das Dateisystem `userpool` erstellt und dessen Eigenschaft `canmount` auf `off` gesetzt. Einhängepunkte für untergeordnete Benutzerdateisysteme werden auf ein einziges Verzeichnis (`/export/home`) gesetzt. Am übergeordneten Dateisystem gesetzte Eigenschaften werden von untergeordneten Dateisystemen geerbt; das übergeordnete Dateisystem selbst wird jedoch nicht eingehängt.

```
# 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
```

Die Eigenschaft `recordsize`

Die Eigenschaft `recordsize` legt eine empfohlene Blockgröße für Dateien im Dateisystem fest.

Diese Eigenschaft dient zur Zusammenarbeit mit Datenbanken, die auf Dateien in festen Blockgrößen zugreifen. ZFS passt Blockgrößen automatisch an interne Algorithmen an, die für typische Zugriffsstrukturen optimiert wurden. Für Datenbanken, die sehr große Dateien erstellen, Dateien jedoch mit wahlfreiem Zugriff in kleineren Blöcken lesen, können diese Algorithmen unter Umständen nicht optimal sein. Wenn Sie die Eigenschaft `recordsize` auf einen Wert setzen, der der Datensatzgröße der betreffenden Datenbank entspricht bzw. größer als diese ist, kann die Leistung bedeutend verbessert werden. Die Verwendung dieser

Eigenschaft für allgemeine Dateisysteme kann sich negativ auf die Leistung auswirken und wird nicht empfohlen. Die angegebene Größe muss ein Zweierpotenzwert sein, der größer als oder gleich 512 Byte und kleiner als oder gleich 128 KB ist. Das Ändern der Eigenschaft `recordsize` eines Dateisystems wirkt sich nur auf Dateien aus, die nach der Änderung erstellt wurden. Bereits vorhandene Dateien bleiben unverändert.

Die Abkürzung der Eigenschaft lautet `recsize`.

Die Eigenschaft `volsize`

Die Eigenschaft `volsize` legt die logische Größe eines Volumes fest. Standardmäßig wird beim Erstellen eines Volumes Speicherplatz der gleichen Kapazität reserviert. Alle Änderungen der Eigenschaft `volsize` werden entsprechend in der Reservierung geändert. Diese Überprüfungen dienen zum Verhindern unerwarteten Systemverhaltens. Je nachdem, wie ein Volume benutzt wird, kann es undefiniertes Systemverhalten oder Datenbeschädigung verursachen, wenn es weniger Speicherplatz enthält, als es eigentlich angefordert hat. Solche Effekte können auch auftreten, wenn die Kapazität eines Volumes geändert wird, während es in Benutzung ist. Dies gilt besonders dann, wenn die Volume-Kapazität verkleinert wird. Gehen Sie bei der Änderung einer Volume-Kapazität stets mit äußerster Sorgfalt vor.

Obwohl dies nicht empfohlen wird, können Sie ein Sparse-Volume erstellen, indem Sie für den Befehl `-zfs create -V` das Flag `s` angeben oder die Reservierung nach der Erstellung des Volumes entsprechend ändern. Bei *Sparse-Volumes* ist der Wert der Reservierung ungleich der Volume-Kapazität. Änderungen der Eigenschaft `volsize` wirken sich bei Sparse-Volumes nicht auf den Reservierungswert aus.

Weitere Informationen zur Verwendung von Volumes finden Sie unter „ZFS-Volumes“ auf Seite 283.

Benutzerdefinierte ZFS-Eigenschaften

Zusätzlich zu den nativen Eigenschaften unterstützt ZFS auch beliebige benutzerdefinierte Eigenschaften. Benutzerdefinierte Eigenschaften wirken sich nicht auf das ZFS-Verhalten aus, können jedoch zum Versehen von Datasets mit Informationen, die für Ihre lokalen Gegebenheiten wichtig sind, verwendet werden.

Namen benutzerdefinierter Eigenschaften müssen den folgenden Konventionen genügen:

- Sie müssen einen Doppelpunkt (:) enthalten, damit sie von nativen Eigenschaften unterschieden werden können.
- Sie dürfen Kleinbuchstaben, Zahlen und die folgenden Interpunktionszeichen enthalten: ':', '+', '.', '_'.
- Der Name einer benutzerdefinierten Eigenschaft darf maximal 256 Zeichen lang sein.

Namen benutzerdefinierter Eigenschaften sollten generell in die folgenden beiden Komponenten aufgeteilt werden, obwohl dieser Namespace für ZFS nicht obligatorisch ist:

module:property

Bei der programmatischen Verwendung von Eigenschaften sollten Sie für die Komponente *Modul* einen umgekehrten DNS-Domänennamen verwenden. Dadurch wird die Wahrscheinlichkeit verringert, dass zwei unabhängig voneinander entwickelte Pakete die gleiche Eigenschaft für unterschiedliche Zwecke nutzen. Eigenschaftsnamen, die mit `com.oracle` beginnen, sind für Oracle Corporation reserviert.

Die Werte benutzerdefinierter Eigenschaften müssen folgenden Konventionen entsprechen:

- Sie müssen aus beliebigen Zeichenketten bestehen, die immer vererbt und niemals validiert werden.
- Der Wert einer benutzerdefinierten Eigenschaft darf maximal 1024 Zeichen lang sein.

Beispiel:

```
# zfs set dept:users=finance userpool/user1
# zfs set dept:users=general userpool/user2
# zfs set dept:users=itops userpool/user3
```

Alle Befehle, die Eigenschaften verwenden (z. B. `zfs list`, `zfs get`, `zfs set` usw.) können native und benutzerdefinierte Eigenschaften nutzen.

Beispiel:

```
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
```

Benutzerdefinierte Eigenschaften können mit dem Befehl `zfs inherit` gelöscht werden. Beispiel:

```
# zfs inherit -r dept:users userpool
```

Wenn die betreffende Eigenschaft nicht in einem übergeordneten Dataset definiert wurde, wird sie komplett entfernt.

Abfragen von ZFS-Dateisysteminformationen

Der Befehl `zfs list` bietet einen umfassenden Mechanismus zum Anzeigen und Abfragen von Dataset-Informationen. In diesem Abschnitt werden grundlegende und komplexere Abfragen erläutert.

Auflisten grundlegender ZFS-Informationen

Mit dem Befehl `zfs list` ohne Optionen können Sie sich grundlegende Dataset-Informationen anzeigen lassen. Dieser Befehl zeigt die Namen aller Datasets im System sowie die Werte der Eigenschaften `used`, `available`, `referenced` und `mountpoint` an. Weitere Informationen zu diesen Eigenschaften finden Sie unter „ZFS-Eigenschaften“ auf Seite 181.

Beispiel:

```
# zfs list
users                   2.00G  64.9G    32K  /users
users/home              2.00G  64.9G    35K  /users/home
users/home/cindy         548K  64.9G   548K  /users/home/cindy
users/home/mark         1.00G  64.9G  1.00G  /users/home/mark
users/home/neil         1.00G  64.9G  1.00G  /users/home/neil
```

Mithilfe dieses Befehls können Sie auch Informationen zu bestimmten Datasets anzeigen. Geben Sie dazu in der Befehlszeile den Namen des gewünschten Datasets an. Darüber hinaus können Sie mit der Option `-r` rekursiv Informationen zu allen untergeordneten Datasets anzeigen. Beispiel:

```
# zfs list -t all -r users/home/mark
NAME                        USED  AVAIL  REFER  MOUNTPOINT
users/home/mark            1.00G  64.9G  1.00G  /users/home/mark
users/home/mark@yesterday      0      -  1.00G  -
users/home/mark@today          0      -  1.00G  -
```

Sie können den Befehl `zfs list` zusammen mit dem Einhängepunkt eines Dateisystems verwenden. Beispiel:

```
# zfs list /user/home/mark
NAME              USED  AVAIL  REFER  MOUNTPOINT
users/home/mark  1.00G  64.9G  1.00G  /users/home/mark
```

Das folgende Beispiel zeigt, wie grundlegende Informationen zum Dateisystem `tank/home/gina` und allen seinen untergeordneten Dateisystemen angezeigt werden können:

```
# zfs list -r users/home/gina
NAME                          USED  AVAIL  REFER  MOUNTPOINT
users/home/gina              2.00G  62.9G    32K  /users/home/gina
users/home/gina/projects     2.00G  62.9G    33K  /users/home/gina/projects
users/home/gina/projects/fs1  1.00G  62.9G  1.00G  /users/home/gina/projects/fs1
users/home/gina/projects/fs2  1.00G  62.9G  1.00G  /users/home/gina/projects/fs2
```

Zusätzliche Informationen zum Befehl `zfs list` finden Sie in der Man Page `zfs(1M)`.

Erstellen komplexer ZFS-Abfragen

Die Ausgabe des Befehls `zfs list` kann mithilfe der Optionen `-o`, `-f` und `-H` speziell angepasst werden.

Mit der Option `-o` und einer kommagetrennten Liste gewünschter Eigenschaften können Sie die Ausgabe von Eigenschaftswerten anpassen. Sie können jede Dataset-Eigenschaft als gültiges Argument angeben. Eine Liste aller unterstützten Dataset-Eigenschaften finden Sie unter „ZFS-Eigenschaften“ auf Seite 181. Zusätzlich zu den hier definierten Eigenschaften kann die Option `-o` auch das Literal `name` enthalten. In diesem Fall enthält die Befehlsausgabe auch den Namen des Datasets.

Im folgenden Beispiel wird mithilfe von `zfs list` der Dataset-Name zusammen mit den Eigenschaftswerten `sharenfs` und `mountpoint` angezeigt.

```
# zfs list -r -o name,sharenfs,mountpoint users/home
NAME                          SHARENFS  MOUNTPOINT
users/home                    on        /users/home
users/home/cindy              on        /users/home/cindy
users/home/gina               on        /users/home/gina
users/home/gina/projects      on        /users/home/gina/projects
users/home/gina/projects/fs1  on        /users/home/gina/projects/fs1
users/home/gina/projects/fs2  on        /users/home/gina/projects/fs2
users/home/mark               on        /users/home/mark
users/home/neil               on        /users/home/neil
```

Mit der Option `-t` können Sie festlegen, welche Dataset-Typen angezeigt werden sollen. Zulässige Typen sind in der folgenden Tabelle aufgeführt.

TABELLE 5–2 Typen von ZFS-Objekten

Typ	Beschreibung
`filesystem`	Dateisysteme und Klone
`volume`	Volumes
`share`	Dateisystemfreigabe
`snapshot`	Schnappschüsse

Die Option `-t` liest eine kommagetrennte Liste der anzuzeigenden Dataset-Typen. Im folgenden Beispiel wird mithilfe der Optionen `-t` und `-o` für alle Dateisysteme gleichzeitig der Name und die Eigenschaft `used` angezeigt:

```
# zfs list -r -t filesystem -o name,used users/home
NAME                           USED
users/home                    4.00G
users/home/cindy               548K
users/home/gina               2.00G
users/home/gina/projects      2.00G
users/home/gina/projects/fs1  1.00G
users/home/gina/projects/fs2  1.00G
users/home/mark               1.00G
users/home/neil               1.00G
```

Mithilfe der Option `-H` kann bei der Ausgabe des Befehls `zfs list` die Titelzeile unterdrückt werden. Bei Verwendung der Option `-H` werden Leerzeichen durch Tabulatorzeichen ersetzt. Diese Option ist bei der Verwendung der Befehlsausgabe für programmatische Anwendungen (z. B. Skripten) nützlich. Das folgende Beispiel zeigt die Ausgabe des Befehls `zfs list` mit der Option `-H`.

```
# zfs list -r -H -o name users/home
users/home
users/home/cindy
users/home/gina
users/home/gina/projects
users/home/gina/projects/fs1
users/home/gina/projects/fs2
users/home/mark
users/home/neil
```

Verwalten von ZFS-Eigenschaften

Dataset-Eigenschaften werden mithilfe der Unterbefehle `set`, `inherit` und `get` des Befehls `zfs` verwaltet.

- „Setzen von ZFS-Eigenschaften“ auf Seite 198
- „Vererben von ZFS-Eigenschaften“ auf Seite 199
- „Abfragen von ZFS-Eigenschaften“ auf Seite 200

Setzen von ZFS-Eigenschaften

Sie können konfigurierbare Dataset-Eigenschaften mit dem Befehl `zfs set` setzen. Bei der Erstellung eines Datasets können Eigenschaften auch mit dem Befehl `zfs create` gesetzt werden. Eine Liste der konfigurierbaren Dataset-Eigenschaften finden Sie unter „Konfigurierbare native ZFS-Eigenschaften“ auf Seite 191.

Der Befehl `zfs set` verwendet ein Eigenschaft-Wert-Paar im Format *Eigenschaft*=*Wert*, dem ein Dataset-Name folgt. Während eines Aufrufs von `zfs set` kann nur eine Eigenschaft gesetzt oder geändert werden.

Im folgenden Beispiel wird die Eigenschaft `atime` von `tank/home` auf `off` gesetzt.

```
# zfs set atime=off tank/home
```

Darüber hinaus können bei der Erstellung eines Dateisystems beliebige Dateisystemeigenschaften gesetzt werden. Beispiel:

```
# zfs create -o atime=off tank/home
```

Spezielle numerische Eigenschaftswerte können durch Verwendung der folgenden verständlichen Suffixe (in ansteigender Größenordnung) angegeben werden: `BKMGTPEZ`. Allen diesen Suffixen außer dem Suffix `B`, das für Byte steht, kann ein `b` (für "Byte”) nachgestellt

werden. In den folgenden vier Beispielen des Befehls `zfs set` werden entsprechende numerische Ausdrücke angegeben, mit denen die Eigenschaft `quota` gesetzt wird. Damit werden Kontingente im Dateisystem `users/home/mark` auf 20 GB gesetzt:

```
# zfs set quota=20G users/home/mark
# zfs set quota=20g users/home/mark
# zfs set quota=20GB users/home/mark
# zfs set quota=20gb users/home/mark
```

Wenn Sie versuchen, eine Eigenschaft in einem Dateisystem festzulegen, das 100 % voll ist, wird eine Meldung wie die Folgende angezeigt.

```
# zfs set quota=20gb users/home/mark
cannot set property for '/users/home/mark': out of space
```

Bei Zeichenkettenwerten wird Groß- und Kleinschreibung unterschieden. Diese Werte dürfen nur Kleinbuchstaben enthalten. Ausnahmen bilden die Werte der Eigenschaften `mountpoint` und `sharenfs`; die Werte dieser Eigenschaften dürfen sowohl Groß- als auch Kleinbuchstaben enthalten.

Weitere Informationen zum Befehl `zfs set` finden Sie in der Man Page `zfs(1M)`.

Vererben von ZFS-Eigenschaften

Alle festlegbaren Eigenschaften, mit Ausnahme der Quota und Reservierungen, übernehmen ihren Wert aus dem übergeordneten Dateisystem, es sei denn, Quota oder Reservierung wird explizit im untergeordneten Dateisystem festgelegt. Wenn das entsprechende übergeordnete Dateisystem für eine vererbte Eigenschaft keinen Wert besitzt, wird der Standardwert für die betreffende Eigenschaft verwendet. Mit dem Befehl `zfs inherit` können Sie Eigenschaftswerte zurücksetzen, was zur Folge hat, dass der vom übergeordneten Dateisystem vererbte Wert verwendet wird.

Im folgenden Beispiel wird mithilfe des Befehls `zfs set` die Komprimierung für das Dateisystem `tank/home/jeff` aktiviert. Danach wird `zfs inherit` verwendet, um die Eigenschaft `compression` zu löschen, wodurch die Eigenschaft den Standardwert (`off`) des erbt. Da weder bei `home` noch bei `tank` der Wert der Eigenschaft `compression` lokal gesetzt wurde, wird der Standardwert verwendet. Wäre bei beiden die Komprimierung aktiviert, würde der Wert des direkten übergeordneten Dateisystems (in diesem Beispiel `home`) verwendet werden.

```
# zfs set compression=on tank/home/jeff
# zfs get -r compression tank/home
NAME                  PROPERTY     VALUE     SOURCE
tank/home             compression  off       default
tank/home/eric        compression  off       default
tank/home/eric@today  compression  -         -
tank/home/jeff        compression  on        local
```

```
# zfs inherit compression tank/home/jeff
# zfs get -r compression tank/home
NAME                  PROPERTY     VALUE     SOURCE
tank/home             compression  off       default
tank/home/eric        compression  off       default
tank/home/eric@today  compression  -         -
tank/home/jeff        compression  off       default
```

Der Unterbefehl `inherit` wird bei Angabe der Option `-r` rekursiv ausgeführt. Im folgenden Beispiel wird durch den Befehl der Wert für die Eigenschaft `compression` von `tank/home` und allen eventuell vorhandenen untergeordneten Dateisystemen geerbt:

```
# zfs inherit -r compression tank/home
```

Hinweis – Beachten Sie, dass die Option `-r` die Eigenschaftswerte aller untergeordneten Dateisysteme zurücksetzt.

Weitere Informationen zum Befehl `zfs inherit` finden Sie in der Man Page `zfs(1M)`.

Abfragen von ZFS-Eigenschaften

Am Einfachsten können Eigenschaftswerte mit dem Befehl `zfs list` abgefragt werden. Weitere Informationen dazu finden Sie unter „Auflisten grundlegender ZFS-Informationen“ auf Seite 196. Für komplexere Abfragen und Skripten sollten Sie den Befehl `zfs get` verwenden, da dieser ausführlichere Informationen in einem anpassbaren Format anzeigt.

Sie können Dataset-Eigenschaften mit dem Befehl `zfs get` abrufen. Das folgende Beispiel zeigt, wie ein Eigenschaftswert eines Datasets abgerufen werden kann:

```
# zfs get checksum tank/ws
NAME            PROPERTY       VALUE                     SOURCE
tank/ws         checksum       on                        default
```

In der vierten Spalte `SOURCE` wird der Ursprung des betreffende Eigenschaftswerts angezeigt. In der folgenden Tabelle werden die möglichen Ursprungswerte erläutert.

TABELLE 5–3 Mögliche SOURCE-Werte (Befehl `zfs get`)

SOURCE-Wert	Beschreibung
`default`	Dieser Eigenschaftswert wurde für dieses Dataset bzw. seine übergeordneten Datasets nie explizit gesetzt. Es wird der Standardwert für diese Eigenschaft verwendet.
`inherited from` *Dataset-Name*	Dieser Eigenschaftswert wurde vom übergeordneten Dataset geerbt, das in *dataset-name* angegeben ist.

TABELLE 5–3 Mögliche SOURCE-Werte (Befehl `zfs get`) *(Fortsetzung)*

SOURCE-Wert	Beschreibung
`local`	Dieser Eigenschaftswert wurde mithilfe von `zfs set` für dieses Dataset explizit gesetzt.
`temporary`	Dieser Eigenschaftswert wurde mithilfe von `zfs mount -o` gesetzt und gilt nur solange, wie das Dateisystem eingehängt ist. Weitere Informationen zu temporären Eigenschaften von Einhängepunkten finden Sie unter „Verwenden temporärer Einhängepunkte“ auf Seite 207.
- (keiner)	Diese Eigenschaft ist schreibgeschützt. Ihr Wert wird von ZFS bereitgestellt.

Sie können alle Dataset-Eigenschaftswerte mit dem speziellen Schlüsselwort `all` abrufen. In den folgenden Beispielen wird das Schlüsselwort `all` verwendet:

Hinweis – Die Eigenschaften `casesensitivity`, `nbmand`, `normalization`, `sharesmb`, `utf8only` und `vscan` sind in Oracle Solaris 10 nicht voll funktionsfähig, da der Oracle Solaris SMB-Service nicht von Oracle Solaris 10 unterstützt wird.

```
# zfs get all tank/home
NAME       PROPERTY              VALUE                  SOURCE
tank/home  type                  filesystem             -
tank/home  creation              Mon Dec  3 13:10 2012  -
tank/home  used                  291K                   -
tank/home  available             58.7G                  -
tank/home  referenced            291K                   -
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               discard                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               5                      -
```

```
tank/home  utf8only              off                    -
tank/home  normalization         none                   -
tank/home  casesensitivity       mixed                  -
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         291K                   -
tank/home  usedbychildren        0                      -
tank/home  usedbyrefreservation  0                      -
tank/home  logbias               latency                default
tank/home  sync                  standard               default
tank/home  rekeydate             -                      default
tank/home  rstchown              on                     default
```

Mit der Option `-s` des Befehls `zfs get` können Sie die anzuzeigenden Eigenschaften nach Ursprungstyp angeben. Diese Option liest eine kommagetrennte Liste der gewünschten Ursprungstypen ein. Es werden nur Eigenschaften des gewünschten Ursprungstyps angezeigt. Zulässige Ursprungstypen sind `local`, `default`, `inherited`, `temporary` und `none`. Das folgende Beispiel zeigt alle Eigenschaften, die in `tank/ws` lokal gesetzt wurden.

```
# zfs get -s local all tank/ws
NAME     PROPERTY              VALUE                  SOURCE
tank/ws  compression           on                     local
```

Alle oben aufgeführten Optionen können zusammen mit der Option `-r` verwendet werden, um die angegebenen Eigenschaften aller untergeordneten Dateisysteme rekursiv anzuzeigen. Im folgenden Beispiel werden alle temporären Eigenschaften aller Dateisysteme in `tank/home` rekursiv angezeigt:

```
# zfs get -r -s temporary all tank/home
NAME             PROPERTY       VALUE                      SOURCE
tank/home          atime          off                      temporary
tank/home/jeff     atime          off                      temporary
tank/home/mark     quota          20G                      temporary
```

Mithilfe des Befehls `zfs get` können Sie Eigenschaftswerte abfragen ohne ein Zieldateisystem anzugeben, was bedeutet, dass alle Pools bzw. Dateisysteme abgefragt werden. Beispiel:

```
# zfs get -s local all
tank/home               atime          off                   local
tank/home/jeff          atime          off                   local
tank/home/mark          quota          20G                   local
```

Weitere Informationen zum Befehl `zfs get` finden Sie in der Man Page `zfs`(1M).

Abfragen von ZFS-Eigenschaften für Skripten

Der Befehl `zfs get` unterstützt die Optionen `-H` und `-o`, die speziell für die Verwendung dieses Befehl in Skripten vorgesehen sind. Sie können die Option `-H` verwenden, um die Kopfzeileninformationen zu unterdrücken und Leerzeichen durch Tabulatorzeichen zu ersetzen. Dadurch können Daten einfach analysiert werden. Sie können die Option `-o` verwenden, um die Ausgabe wie folgt anzupassen:

- Das Literal `name` kann zusammen mit einer kommagetrennten Liste von Eigenschaften verwendet werden (siehe Abschnitt „ZFS-Eigenschaften“ auf Seite 181.
- Eine kommagetrennte Liste von Literalfeldern, `name`, `value`, `property` und `source`, wird ausgegeben, der ein Leerzeichen und ein Argument folgt. Diese Liste ist eine kommagetrennte Liste von Eigenschaften.

Das folgende Beispiel zeigt, wie mithilfe der Optionen `-H` und `-o` des Befehls `zfs get` ein einzelner Wert abgerufen werden kann.

```
# zfs get -H -o value compression tank/home
on
```

Die Option `-p` gibt numerische Werte exakt aus. 1 MB wird beispielsweise als 1000000 ausgegeben. Diese Option lässt sich wie folgt verwenden:

```
# zfs get -H -o value -p used tank/home
182983742
```

Mit der Option `-r` und allen der o. g. Optionen können Sie Werte für alle untergeordneten Datasets rekursiv abrufen. Im folgenden Beispiel werden die Optionen `-H`, `-o` und `-r` verwendet, um den Dateisystemnamen sowie den Wert der Eigenschaft `used` für `export/home` und die untergeordneten Objekte abzurufen, während die Kopfzeile der Befehlsausgabe unterdrückt wird:

```
# zfs get -H -o name,value -r used export/home
```

Einhängen von ZFS-Dateisystemen

In diesem Abschnitt wird beschrieben, wie ZFS Dateisysteme einhängt.

Verwalten von ZFS-Einhängepunkten

Ein ZFS-Dateisystem wird automatisch eingehängt, wenn es erstellt wird. In diesem Abschnitt wird beschrieben, wie Sie das Verhalten eines Einhängepunkts für ein Dateisystem bestimmen können.

Sie können den Standard-Einhängepunkt für das Dateisystem eines Pools bei dessen Erstellung auch mithilfe der Option `m` von `-zpool create` setzen. Weitere Informationen zum Erstellen von Pools finden Sie unter „Erstellen von ZFS-Speicherpools“ auf Seite 52.

Alle ZFS-Dateisysteme werden beim Systemstart von ZFS mithilfe des SMF-Service (Service Management Facility) `svc://system/filesystem/local` eingehängt. Dateisysteme werden unter */path* eingehängt, wobei *path* den Namen des Dateisystems bezeichnet.

Sie können den Standard-Einhängepunkt überschreiben, indem Sie den Befehl `zfs set` verwenden, um die Eigenschaft `mountpoint` auf einen spezifischen Pfad zu setzen. ZFS erstellt den angegebenen Einhängepunkt bei Bedarf automatisch und hängt das entsprechende Dateisystem automatisch ein.

ZFS-Dateisysteme werden beim Systemstart automatisch eingehängt, ohne dass Sie die Datei `/etc/vfstab` bearbeiten müssen.

Die Eigenschaft `mountpoint` wird vererbt. Wenn die Eigenschaft `mountpoint` von `pool/home` beispielsweise auf `/export/stuff` gesetzt ist, erbt `pool/home/user` für seinen Eigenschaftswert `mountpoint` den Wert `/export/stuff/user`.

Um zu verhindern, dass ein Dateisystem eingehängt wird, setzen Sie die Eigenschaft `mountpoint` auf `none`. Außerdem kann mit der Eigenschaft `canmount` bestimmt werden, ob ein Dateisystem eingehängt werden kann. Weitere Informationen zur Eigenschaft `canmount` finden Sie unter „Die Eigenschaft `canmount`“ auf Seite 193.

Dateisysteme können auch explizit mithilfe von Legacy-Einhängesystemen verwaltet werden, indem `zfs set` verwendet wird, um die Eigenschaft `mountpoint` auf `legacy` zu setzen. Dadurch wird verhindert, dass ZFS ein Dateisystem automatisch einhängt und verwaltet. Stattdessen müssen Sie Legacy-Serviceprogramme wie die Befehle `mount` und `umount` und die Datei `/etc/vfstab` verwenden. Weitere Informationen zu Legacy-Einhängepunkten finden Sie unter „Legacy-Einhängepunkte“ auf Seite 205.

Automatische Einhängepunkte

- Wenn Sie die Eigenschaft `mountpoint` von `legacy` oder `none` auf einen bestimmten Pfad umsetzen, hängt ZFS das betreffende Dateisystem automatisch ein.
- Wenn ZFS ein Dateisystem verwaltet, dieses aber ausgehängt ist und die Eigenschaft `mountpoint` geändert wird, bleibt das Dateisystem weiterhin ausgehängt.

Dateisysteme, deren Eigenschaft `mountpoint` nicht auf `legacy` gesetzt ist, werden von ZFS verwaltet. Im folgenden Beispiel wird ein Dateisystem erstellt, dessen Einhängepunkt automatisch von ZFS verwaltet wird:

```
# 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                        -
```

Sie können die Eigenschaft `mountpoint` auch explizit setzen (siehe folgendes Beispiel):

```
# 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                        -
```

Beim Ändern der Eigenschaft `mountpoint` wird das betreffende Dateisystem automatisch aus dem alten Einhängepunkt ausgehängt und in den neuen Einhängepunkt eingehängt. Einhängepunktverzeichnisse werden je nach Bedarf erstellt. Wenn ZFS ein Dateisystem nicht aushängen kann, weil es noch aktiv ist, wird ein Fehler gemeldet, und das Aushängen muss manuell erzwungen werden.

Legacy-Einhängepunkte

Sie können ZFS-Dateisysteme mit Legacy-Serviceprogrammen verwalten, indem Sie die Eigenschaft `mountpoint` auf `legacy` setzen. Legacy-Dateisysteme müssen mithilfe der Befehle `mount` und `umount` sowie der Datei `/etc/vfstab` verwaltet werden. ZFS hängt Legacy-Dateisysteme beim Systemstart nicht automatisch ein, und die ZFS-Befehle `mount` und `umount` funktionieren mit Dateisystemen dieses Typs nicht. Die folgenden Beispiele zeigen die Einrichtung und Administration eines ZFS-Dateisystems im Legacy-Modus:

```
# zfs set mountpoint=legacy tank/home/eric
# mount -F zfs tank/home/eschrock /mnt
```

Damit Legacy-Dateisysteme beim Systemstart automatisch eingehängt werden, müssen Sie zur Datei `/etc/vfstab` die entsprechenden Einträge hinzufügen. Das folgende Beispiel zeigt, wie der Eintrag in der Datei `/etc/vfstab` aussehen kann:

```
#device         device        mount           FS      fsck    mount   mount
#to mount       to fsck       point           type    pass    at boot options
#

tank/home/eric  -        /mnt           zfs        -        yes        -
```

Die Einträge `device to fsck` und `fsck pass` werden auf - gesetzt, weil der Befehl `fsck` nicht auf ZFS-Dateisysteme anwendbar ist. Weitere Informationen zur ZFS-Datenintegrität finden Sie unter „Transaktionale Semantik“ auf Seite 27.

Einhängen von ZFS-Dateisystemen

ZFS hängt Dateisysteme beim Erstellen dieser Dateisysteme bzw. beim Systemstart automatisch ein. Der Befehl `zfs mount` muss nur verwendet werden, wenn Einhängeoptionen geändert oder Dateisysteme explizit ein- oder ausgehängt werden müssen.

Beim Aufrufen des Befehls `zfs mount` ohne Argumente werden alle von ZFS verwalteten und gegenwärtig eingehängten Dateisysteme angezeigt. Legacy-Einhängepunkte werden nicht angezeigt. Beispiel:

```
# zfs mount | grep tank/home
zfs mount | grep tank/home
tank/home                       /tank/home
tank/home/jeff                  /tank/home/jeff
```

Mit der Option `-a` können Sie alle von ZFS verwalteten Dateisysteme einhängen. Legacy-Dateisysteme werden nicht eingehängt. Beispiel:

```
# zfs mount -a
```

Standardmäßig erlaubt ZFS das Einhängen in ein nicht leeres Verzeichnis nicht. Beispiel:

```
# zfs mount tank/home/lori
cannot mount 'tank/home/lori': filesystem already mounted
```

Legacy-Einhängepunkte müssen mit Legacy-Serviceprogrammen verwaltet werden. Wenn Sie versuchen, dafür ZFS-Befehle zu verwenden, wird ein Fehler ausgegeben. Beispiel:

```
# zfs mount tank/home/bill
cannot mount 'tank/home/bill': legacy mountpoint
use mount(1M) to mount this filesystem
# mount -F zfs tank/home/billm
```

Beim Einhängen eines Dateisystems werden verschiedene Einhängeoptionen verwendet, die auf den zum Dateisystem gehörenden Eigenschaftswerten beruhen. Zwischen Eigenschaften und Einhängeoptionen besteht der folgende Zusammenhang:

TABELLE 5–4 Eigenschaften von ZFS-Einhängepunkten und Einhängeoptionen

Eigenschaft	Einhängeoption
`atime`	`atime/noatime`
`devices`	`devices/nodevices`

TABELLE 5–4 Eigenschaften von ZFS-Einhängepunkten und Einhängeoptionen *(Fortsetzung)*

Eigenschaft	Einhängeoption
exec	exec/noexec
nbmand	nbmand/nonbmand
readonly	ro/rw
setuid	setuid/nosetuid
xattr	xattr/noaxttr

Die Einhängeoption `nosuid` ist ein Alias-Name für `nodevices,nosetuid`.

Verwenden temporärer Einhängepunkte

Wenn die im vorherigen Abschnitt beschriebenen Optionen explizit mithilfe der Option `-o` des Befehls `zfs mount` gesetzt werden, wird der zugehörige Eigenschaftswert temporär überschrieben. Diese Eigenschaftswerte werden vom Befehl `zfs get` mit dem Wert `temporary` gemeldet und beim Aushängen des betreffenden Dateisystems auf ihre ursprünglichen Werte zurückgesetzt. Beim Ändern eines Eigenschaftswerts während des Einhängens eines Dateisystems wird diese Änderung sofort übernommen und überschreibt eventuelle temporäre Einstellungen.

Im folgenden Beispiel wird am Dateisystem `tank/home/neil` die Einhängeoption "schreibgeschützt" temporär gesetzt. Es wird angenommen, dass das Dateisystem ausgehängt ist.

```
# zfs mount -o ro users/home/neil
```

Zum temporären Ändern eines Eigenschaftswerts eines gegenwärtig eingehängten Dateisystems dient die spezielle Option `remount`. Im folgenden Beispiel wird die Eigenschaft `atime` für ein eingehängtes Dateisystem temporär auf `off` gesetzt.

```
# zfs mount -o remount,noatime users/home/neil
NAME             PROPERTY  VALUE  SOURCE
users/home/neil  atime     off    temporary
# zfs get atime users/home/perrin
```

Weitere Informationen zum Befehl `zfs mount` finden Sie in der Man Page `zfs(1M)`.

Aushängen von ZFS-Dateisystemen

ZFS-Dateisysteme können mit dem Befehl `zfs unmount` ausgehängt werden. Der Befehl `unmount` akzeptiert als Argument entweder den Einhängepunkt oder den Dateisystemnamen.

Im folgenden Beispiel wird ein Dateisystem nach seinem Namen ausgehängt:

```
# zfs unmount users/home/mark
```

Im folgenden Beispiel wird ein Dateisystem nach seinem Einhängepunkt ausgehängt:

```
# zfs unmount /users/home/mark
```

Der Befehl `unmount` schlägt fehl, wenn das betreffende Dateisystem aktiv ist. Ein Aushängen eines Dateisystems kann mit der Option `-f` erzwungen werden. Gehen Sie beim erzwungenen Aushängen eines Dateisystems äußerst sorgsam vor, wenn der Inhalt dieses Dateisystems noch verwendet wird. da unvorhergesehenes Anwendungsverhalten die Folge davon sein kann.

```
# zfs unmount tank/home/eric
cannot unmount '/tank/home/eric': Device busy
# zfs unmount -f tank/home/eric
```

Zum Zweck der Abwärtskompatibilität kann der Legacy-Befehl `umount` auch zum Aushängen von ZFS-Dateisystemen verwendet werden. Beispiel:

```
# umount /tank/home/bob
```

Weitere Informationen zum Befehl `zfs unmount` finden Sie in der Man Page `zfs(1M)`.

Freigeben und Sperren von ZFS-Dateisystemen

ZFS kann Dateisysteme durch entsprechendes Setzen der Eigenschaft `sharenfs` automatisch für den Netzwerkzugriff freigeben. Mit dieser Eigenschaft müssen Sie die Datei `/etc/dfs/dfstab` nicht ändern, wenn ein neues Dateisystem freigegeben wurde. Die Eigenschaft `sharenfs` ist eine kommagetrennte Liste mit Optionen, die an den Befehl `share` übergeben wird. Der spezielle Wert `on` ist ein Aliasname für die Standardoptionen der Netzwerkfreigabe, der allen Benutzern Lese- und Schreibberechtigung gewährt. Der Wert `off` gibt an, dass das Dateisystem nicht von ZFS verwaltet wird und über herkömmliche Mittel wie z. B. die Datei `/etc/dfs/dfstab` für den Netzwerkzugriff freigegeben werden kann. Alle Dateisysteme, deren Eigenschaft `sharenfs` nicht auf `off` gesetzt ist, werden beim Systemstart freigegeben.

Einstellen der Freigabesemantik

Standardmäßig sind alle Dateisysteme für den Netzwerkzugriff gesperrt. Zum Freigeben eines neuen Dateisystems für den Netzwerkzugriff müssen Sie den Befehl `zfs set` mit der folgenden Syntax verwenden:

```
# zfs set sharenfs=on tank/home/eric
```

Die Eigenschaft `sharenfs` wird vererbt, und alle Dateisysteme werden bei der Erstellung automatisch für den Netzwerkzugriff freigegeben, wenn deren vererbte Eigenschaft nicht auf `off` gesetzt ist. Beispiel:

```
# zfs set sharenfs=on tank/home
# zfs create tank/home/bill
# zfs create tank/home/mark
# zfs set sharenfs=ro tank/home/bob
```

Die Dateisysteme `tank/home/bill` und `tank/home/mark` sind anfänglich mit Schreibzugriff freigegeben, da sie die Eigenschaft `sharenfs` von `tank/home` erben. Nach dem Setzen dieser Eigenschaft auf `ro` (schreibgeschützt) wird das Dateisystem `tank/home/mark` unabhängig davon, welchen Wert die Eigenschaft `sharenfs` für `tank/home` hat, schreibgeschützt freigegeben.

Sperren von ZFS-Dateisystemen für den Netzwerkzugriff

Obwohl die meisten Dateisysteme beim Systemstart, der Erstellung und beim Löschen automatisch für den Netzwerkzugriff freigegeben bzw. gesperrt werden, kann es manchmal vorkommen, dass Dateisysteme explizit für den Netzwerkzugriff gesperrt werden müssen. Dazu dient der Befehl `zfs unshare`. Beispiel:

```
# zfs unshare tank/home/mark
```

Dieser Befehl sperrt das Dateisystem `tank/home/mark` für den Netzwerkzugriff. Zum Sperren aller ZFS-Dateisysteme auf einem System benötigen Sie die Option `-a`.

```
# zfs unshare -a
```

Freigeben von ZFS-Dateisystemen für den Netzwerkzugriff

In den meisten Fällen reicht das automatische ZFS-Verhalten des Freigebens von Dateisystemen beim Systemstart und Erstellen eines Dateisystems für den Normalbetrieb aus. Falls Dateisysteme doch einmal für den Netzwerkzugriff gesperrt werden müssen, können Sie diese mit dem Befehl `zfs share` wieder freigeben. Beispiel:

```
# zfs share tank/home/mark
```

Sie können auch alle ZFS-Dateisysteme auf einem System mithilfe der Option `-a` freigeben.

```
# zfs share -a
```

Freigabeverhalten bei Legacy-Dateisystemen

Wenn die Eigenschaft `sharenfs` auf `off` gesetzt ist, versucht ZFS niemals, das betreffende Dateisystem für den Netzwerkzugriff freizugeben bzw. zu sperren. Dieser Wert ermöglicht die FreigabeAdministration mithilfe herkömmlicher Mittel wie z. B. der Datei `/etc/dfs/dfstab`.

Im Gegensatz zum Befehl `mount` funktionieren die Befehle `share` und `unshare` auch für ZFS-Dateisysteme. Deswegen können Sie ein Dateisystem manuell mit Optionen freigeben, die sich von den Optionen der Eigenschaft `sharenfs` unterscheiden. Davon wird jedoch abgeraten.

Sie sollten NFS-Freigaben entweder vollständig von ZFS oder vollständig mithilfe der Datei `/etc/dfs/dfstab` verwalten lassen. Das administrative ZFS-Modell ist einfacher und nicht so aufwändig wie das traditionelle Modell.

Einstellen von ZFS-Kontingenten und -Reservierungen

Mit der Eigenschaft `quota` können Sie die Festplattenkapazität, die ein Dateisystem verwenden kann, beschränken. Darüber hinaus können Sie mit der Eigenschaft `reservation` für ein Dateisystem verfügbare Festplattenkapazität garantieren. Beide Eigenschaften gelten für das Dateisystem, in dem sie festgelegt werden, und für alle untergeordneten Dateisysteme dieses Dateisystems.

Wenn für das Dateisystem `tank/home` ein Kontingent festgelegt wurde, heißt das, dass die insgesamt von `tank/home` *und allen seinen untergeordneten Dateisystemen* belegte Festplattenkapazität dieses Kontingent nicht überschreiten kann. Genauso wird beim Festlegen einer Reservierung für `tank/home` diesem Dateisystem *und allen seinen untergeordneten Dateisystemen* dieser Speicherplatz garantiert. Die von einem Dateisystem und allen seinen untergeordneten Dateisystemen belegte Festplattenkapazität wird von der Eigenschaft `used` verfolgt.

Die Eigenschaften `refquota` und `refreservation` stehen zur Administration von Systemspeicherplatz zur Verfügung. Die von untergeordneten Objekten wie Schnappschüssen und Klonen beanspruchte Festplattenkapazität wird dabei nicht berücksichtigt.

In diesem Solaris-Release können Sie ein *user*- oder *group*-Kontingent für die Festplattenkapazität festlegen, die von Dateien beansprucht wird, die zu einem bestimmten Benutzer oder einer bestimmten Gruppe gehören. Die Kontingenteigenschaften des Benutzers oder der Gruppe können nicht auf einem Volume, einem Dateisystem vor Dateisystem-Version 4 oder einem Pool vor Pool-Version 15 eingerichtet werden.

Beachten Sie die folgenden Faktoren, um festzustellen, welche Kontingent- und Reservierungsfunktionen sich am besten für die Administration Ihrer Dateisysteme anbieten:

- Die Eigenschaften `quota` und `reservation` eignen sich zur Administration von Festplattenkapazität, die durch Dateisysteme und deren untergeordnete Dateisysteme belegt ist.
- Die Eigenschaften `refquota` und `refreservation` eignen sich zur Administration von Festplattenkapazität, die durch Dateisysteme belegt ist.
- Das Setzen der Eigenschaft `refquota` oder `refreservation` auf einen höheren Wert als die Eigenschaft `quota` oder `reservation` hat keine Wirkung. Wenn Sie die Eigenschaft `quota` oder `refquota` setzen, schlagen Vorgänge, bei denen einer dieser Werte überschritten wird, fehl. Es ist möglich, ein Kontingent (`quota`) zu überschreiten, das größer ist als `refquota`. Wenn Schnappschussblöcke beispielsweise geändert werden, kann tatsächlich eher `quota` als `refquota` überschritten werden.

- Benutzer- und Gruppenkontingente vereinfachen die Administration des Festplattenspeichers bei Vorhandensein vieler Benutzerkonten, zum Beispiel in einer universitären Umgebung.

Weitere Informationen zum Einrichten von Kontingenten und Reservierungen finden Sie unter „Setzen von Kontingenten für ZFS-Dateisysteme“ auf Seite 211und „Setzen von Reservierungen für ZFS-Dateisysteme“ auf Seite 214.

Setzen von Kontingenten für ZFS-Dateisysteme

Kontingente für ZFS-Dateisysteme können mit den Befehlen `zfs set` und `zfs get` festgelegt und angezeigt werden. Im folgenden Beispiel wird für das Dateisystem `tank/home/jeff`ein Kontingent von 10 GB festgelegt:

```
# zfs set quota=10G tank/home/jeff
# zfs get quota tank/home/jeff
NAME            PROPERTY  VALUE  SOURCE
tank/home/jeff  quota     10G    local
```

ZFS-Kontingente wirken sich auch auf die Ausgabe der Befehle `list` und `df` aus. Beispiel:

```
# zfs list -r tank/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home          1.45M  66.9G    36K  /tank/home
tank/home/eric      547K  66.9G   547K  /tank/home/eric
tank/home/jeff      322K  10.0G   291K  /tank/home/jeff
tank/home/jeff/ws    31K  10.0G    31K  /tank/home/jeff/ws
tank/home/lori      547K  66.9G   547K  /tank/home/lori
tank/home/mark       31K  66.9G    31K  /tank/home/mark
# df -h /tank/home/jeff
Filesystem             Size  Used Avail Use% Mounted on
tank/home/jeff          10G  306K   10G   1% /tank/home/jeff
```

Bitte beachten Sie, dass obwohl für `tank/home` 66,9 GB Festplattenkapazität verfügbar sind, wegen des für `tank/home/jeff` festgelegten Kontingents für `tank/home/jeff` und `tank/home/jeff/ws` nur 10 GB verfügbar sind.

Kontingente können nicht auf Werte gesetzt werden, die kleiner als der gegenwärtig vom betreffenden Dateisystem belegte Speicherplatz sind. Beispiel:

```
# zfs set quota=10K tank/home/jeff
cannot set property for 'tank/home/jeff':
size is less than current used or reserved space
```

Sie können eine `refquota` für ein Dateisystem festlegen, die den Speicherplatz begrenzt, den das Dateisystem belegen kann. Dieser absolute Grenzwert berücksichtigt keine Festplattenkapazität, die von untergeordneten Objekten belegt wird. Beispielsweise wirkt sich die von Schnappschüssen beanspruchte Festplattenkapazität nicht auf das 10-GB-Kontingent von studentA aus.

```
# zfs set refquota=10g students/studentA
# zfs list -t all -r students
NAME                          USED  AVAIL  REFER  MOUNTPOINT
students                      150M  66.8G    32K  /students
students/studentA             150M  9.85G   150M  /students/studentA
students/studentA@yesterday      0      -   150M  -
# zfs snapshot students/studentA@today
# zfs list -t all -r students
NAME                          USED  AVAIL  REFER  MOUNTPOINT
students                      150M  66.8G    32K  /students
students/studentA             150M  9.90G   100M  /students/studentA
students/studentA@yesterday  50.0M      -   150M  -
students/studentA@today          0      -   100M  -
```

Sie können ein weiteres Kontingent für ein Dateisystem festlegen, um die Administration der durch Schnappschüsse belegten Festplattenkapazität zu erleichtern. Beispiel:

```
# zfs set quota=20g students/studentA
# zfs list -t all -r students
NAME                          USED  AVAIL  REFER  MOUNTPOINT
students                      150M  66.8G    32K  /students
students/studentA             150M  9.90G   100M  /students/studentA
students/studentA@yesterday  50.0M      -   150M  -
students/studentA@today          0      -   100M  -
```

In diesem Szenario kann `studentA` den mit "`refquota`" festgelegten absoluten Grenzwert (10 GB) erreichen und selbst bei vorhandenen Schnappschüssen wiederherzustellende Dateien entfernen.

Im obigen Beispiel wird das kleinere der beiden Kontingente (10 GB im Vergleich mit 20 GB) in der Ausgabe von `zfs list` angezeigt. Zum Anzeigen der Werte beider Kontingente verwenden Sie den Befehl `zfs get`. Beispiel:

```
# zfs get refquota,quota students/studentA
NAME               PROPERTY  VALUE              SOURCE
students/studentA  refquota  10G                local
students/studentA  quota     20G                local
```

Einrichten von Benutzer- und Gruppenkontingenten auf einem ZFS-Dateisystem

Unter Verwendung des Befehls `zfs userquota` oder `zfs groupquota` können Sie ein Benutzer- oder Gruppenkontingent wie folgt einrichten. Beispiel:

```
# zfs create students/compsci
# zfs set userquota@student1=10G students/compsci
# zfs create students/labstaff
# zfs set groupquota@labstaff=20GB students/labstaff
```

Zeigen Sie das aktuelle Benutzer- oder Gruppenkontingent wie folgt an:

```
# zfs get userquota@student1 students/compsci
NAME              PROPERTY            VALUE               SOURCE
students/compsci  userquota@student1  10G                 local
```

```
# zfs get groupquota@labstaff students/labstaff
NAME                PROPERTY             VALUE                SOURCE
students/labstaff   groupquota@labstaff  20G                  local
```

Sie können die von allgemeinen Benutzern und Gruppen belegte Festplattenkapazität durch Abfrage der folgenden Eigenschaften anzeigen:

```
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      350M   none
POSIX User  student1  426M    10G
# zfs groupspace students/labstaff
TYPE         NAME      USED  QUOTA
POSIX Group  labstaff  250M    20G
POSIX Group  root      350M   none
```

Um die von einzelnen Benutzern oder Gruppen belegte Festplattenkapazität zu ermitteln, fragen Sie die folgenden Eigenschaften ab:

```
# zfs get userused@student1 students/compsci
NAME              PROPERTY           VALUE              SOURCE
students/compsci  userused@student1  550M               local
# zfs get groupused@labstaff students/labstaff
NAME                PROPERTY            VALUE               SOURCE
students/labstaff   groupused@labstaff  250                 local
```

Die Eigenschaften der Benutzer- und Gruppenkontingente werden nicht über den Befehl `zfs get all` *dataset* angezeigt. Dieser führt die Eigenschaften aller anderen Dateisysteme auf.

Sie können ein Benutzer- oder Gruppenkontingent folgendermaßen entfernen:

```
# zfs set userquota@student1=none students/compsci
# zfs set groupquota@labstaff=none students/labstaff
```

ZFS-Benutzer- und Gruppenkontingente in ZFS-Dateisystemen besitzen folgende Merkmale:

- Ein in einem übergeordneten Dateisystem eingerichtetes Benutzer- oder Gruppenkontingent wird nicht automatisch von einem untergeordneten Dateisystem übernommen.
- Allerdings wird das Benutzer- oder Gruppenkontingent angewendet, wenn ein Klon oder Schnappschuss aus einem Dateisystem mit einem Benutzer- oder Gruppenkontingent erstellt wird. Ebenso wird ein Benutzer- oder Gruppenkontingent in das Dateisystem aufgenommen, wenn mit dem Befehl `zfs send` ein Datenstrom erstellt wird, auch ohne Anwendung der Option `-R`.
- Benutzer ohne Privilegien haben nur Zugriff auf ihre eigene Festplattenkapazität. Der Root-Benutzer oder ein Benutzer, dem das Privileg `userused` oder `groupused` erteilt wurde, kann auf Informationen zur Berechnung der Festplattenkapazität sämtlicher Benutzer und Gruppen zugreifen.
- Die Eigenschaften `userquota` und `groupquota` können nicht auf ZFS-Volumes, auf einem Dateisystem vor Dateisystem-Version 4 oder auf einem Pool vor Pool-Version 15 eingerichtet werden.

Das Inkrafttreten von Benutzer- und Gruppenkontingenten kann einige Sekunden dauern. Die Verzögerung kann bedeuten, dass Benutzer ihr Kontingent überschreiten, bevor das System dies registriert und weitere Schreibvorgänge mit der Fehlermeldung `EDQUOT` zurückweist.

Mit dem Legacy-Befehl `quota` können Sie Benutzerkontingente in einer NFS-Umgebung überprüfen, zum Beispiel beim Einhängen eines ZFS-Dateisystems. Ohne Auswahl von Optionen zeigt der Befehl `quota` nur Ergebnisse, wenn das Benutzerkontingent überschritten wird. Beispiel:

```
# zfs set userquota@student1=10m students/compsci
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      350M   none
POSIX User  student1  550M    10M
# quota student1
Block limit reached on /students/compsci
```

Wenn Sie das Benutzerkontingent zurücksetzen und die Begrenzung des Kontingents nicht länger überschritten ist, können Sie zur Überprüfung des Benutzerkontingents den Befehl `quota -v` verwenden. Beispiel:

```
# zfs set userquota@student1=10GB students/compsci
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      350M   none
POSIX User  student1  550M    10G
# quota student1
# quota -v student1
Disk quotas for student1 (uid 102):
Filesystem     usage  quota  limit    timeleft  files  quota  limit    timeleft
/students/compsci
              563287 10485760 10485760           -      -      -      -           -
```

Setzen von Reservierungen für ZFS-Dateisysteme

Unter einer ZFS-*Reservierung* versteht man eine Zuweisung von Festplattenkapazität aus dem Pool, die einem Dataset garantiert ist. Sie können keine Reservierungen vornehmen, wenn die angeforderte Festplattenkapazität im Pool nicht zur Verfügung steht. Der Gesamtbetrag aller noch ausstehenden und nicht belegten Reservierungen darf die Gesamtsumme der nicht belegten Festplattenkapazität im Pool nicht überschreiten. ZFS-Reservierungen können mit den Befehlen `zfs set` und `zfs get` festgelegt und angezeigt werden. Beispiel:

```
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
NAME            PROPERTY     VALUE   SOURCE
tank/home/bill  reservation  5G      local
```

ZFS-Reservierungen können sich auf die Ausgabe des Befehls `zfs list` auswirken. Beispiel:

```
# zfs list -r tank/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home          5.00G  61.9G    37K  /tank/home
tank/home/bill       31K  66.9G    31K  /tank/home/bill
tank/home/jeff      337K  10.0G   306K  /tank/home/jeff
tank/home/lori      547K  61.9G   547K  /tank/home/lori
tank/home/mark       31K  61.9G    31K  /tank/home/mark
```

Bitte beachten Sie, dass `tank/home` 5 GB Festplattenkapazität belegt, obwohl `tank/home` und seinen untergeordneten Dateisysteme tatsächlich viel weniger als 5 GB Festplattenkapazität belegen. Der belegte Speicherplatz berücksichtigt die Reservierung für `tank/home/bill`. Reservierungen werden in die belegten Festplattenkapazität des übergeordneten Dateisystems einbezogen und auf sein Kontingent und/oder seine Reservierung angerechnet.

```
# 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
```

Ein Dataset kann mehr Festplattenkapazität belegen, als für seine Reservierung vorgesehen ist, solange nicht reservierter Speicherplatz im Pool vorhanden ist und der aktuell vom Dataset belegte Speicherplatz unter seinem Kontingent liegt. Ein Dataset kann keine Festplattenkapazität belegen, die für ein anderes Dataset reserviert wurde.

Reservierungen sind nicht kumulativ. Das bedeutet, dass durch einen zweiten Aufruf von `zfs set` zum Setzen einer Reservierung der Speicherplatz der zweiten Reservierung nicht zum Speicherplatz der vorhandenen Reservierung addiert wird. Die zweite Reservierung ersetzt stattdessen die erste Reservierung. Beispiel:

```
# zfs set reservation=10G tank/home/bill
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
NAME            PROPERTY     VALUE   SOURCE
tank/home/bill  reservation  5G      local
```

Durch Setzen der Reservierung `refreservation` können Sie einem Dataset Festplattenkapazität garantieren, in der die von Schnappschüssen und Klonen belegte Festplattenkapazität nicht berücksichtigt ist. Diese Reservierung wird in die Berechnung der Speicherplatzkapazität für das diesem Dataset übergeordnete Dataset einbezogen und auf die Kontingente und Reservierung für das übergeordnete Dataset angerechnet. Beispiel:

```
# 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
```

Sie können außerdem für dasselbe Dataset eine Reservierung festlegen, um Speicherplatz für das Dataset und für Schnappschüsse zu garantieren. Beispiel:

```
# 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
```

Normale Reservierungen werden in die Berechnung des belegten Speicherplatzes durch das übergeordnete Dataset einbezogen.

Im obigen Beispiel wird das kleinere der beiden Kontingente (10 GB im Vergleich mit 20 GB) in der Ausgabe von `zfs list` angezeigt. Zum Anzeigen der Werte beider Kontingente verwenden Sie den Befehl `zfs get`. Beispiel:

```
# zfs get reservation,refreserv profs/prof1
NAME         PROPERTY        VALUE        SOURCE
profs/prof1  reservation     20G          local
profs/prof1  refreservation  10G          local
```

Wenn `refreservation` gesetzt ist, wird ein Schnappschuss nur zugelassen, wenn außerhalb dieser Reservierung genügend nicht reservierter Speicherplatz im Pool vorhanden ist, um die Menge der aktuell *referenzierten* Byte im Dataset aufzunehmen.

Aktualisieren von ZFS-Dateisystemen

Wenn Sie über ZFS-Dateisysteme aus einer früheren Solaris-Version verfügen, können Sie diese mit dem Befehl `zfs upgrade` aktualisieren, um die Dateisystemfunktionen in der aktuellen Version nutzen zu können. Darüber hinaus werden Sie von diesem Befehl darauf hingewiesen, wenn Ihre Dateisysteme mit älteren Versionen laufen.

Dieses Dateisystem läuft beispielsweise mit der aktuellen Version 5.

```
# zfs upgrade
This system is currently running ZFS filesystem version 5.

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

Verwenden Sie diesen Befehl, um die mit den einzelnen Dateisystemversionen verfügbaren Funktionen zu ermitteln.

```
# zfs upgrade -v
The following filesystem versions are supported:

VER  DESCRIPTION
---  --------------------------------------------------------
 1   Initial ZFS filesystem version
 2   Enhanced directory entries
 3   Case insensitive and File system unique identifier (FUID)
 4   userquota, groupquota properties
 5   System attributes

For more information on a particular version, including supported releases,
see the ZFS Administration Guide.
```

◆ ◆ ◆ KAPITEL 6

Arbeiten mit Oracle Solaris ZFS-Schnappschüssen und -Klonen

Dieses Kapitel enthält Informationen zum Erstellen und Verwalten von Oracle Solaris ZFS-Schnappschüssen und -Klonen. Außerdem finden Sie hier auch Informationen zum Speichern von Schnappschüssen.

Dieses Kapitel enthält die folgenden Abschnitte:

Überblick über ZFS-Schnappschüsse

Ein *Schnappschuss* ist eine schreibgeschützte Kopie eines Dateisystems bzw. Volume. Schnappschüsse können praktisch sofort erstellt werden und verbrauchen keinen zusätzlichen Festplattenspeicherplatz innerhalb des Pools. Wenn sich die Daten innerhalb des aktiven Datensatzes ändern, verbraucht der Schnappschuss Festplattenspeicher, da er auf die veralteten Daten verweist und daher kein Speicherplatz auf der Festplatte freigegeben wird.

ZFS-Schnappschüsse besitzen die folgenden Leistungsmerkmale:

- Sie bleiben auch nach Systemneustarts wirksam.
- Die theoretische Maximalanzahl möglicher Schnappschüsse beträgt 2^{64}.
- Schnappschüsse verwenden keine separaten Zusatzspeicher. Schnappschüsse belegen Festplattenkapazität direkt im gleichen Speicher-Pool, wie auch das Dateisystem oder Volume, aus dem sie erstellt wurden.

- Rekursive Schnappschüsse werden schnell in einem unteilbaren Vorgang erstellt. Schnappschüsse werden entweder zusammen (d. h. alle auf einmal) oder gar nicht erstellt. Der Vorteil solcher unteilbarer Schnappschüsse besteht darin, dass die Schnappschussdaten auch bei Dateisystemhierarchien zu einem einzigen konsistenten Zeitpunkt erstellt werden.

Auf Schnappschüsse von Volumes kann nicht direkt zugegriffen werden, aber sie können geklont, als Sicherungskopie gesichert und wiederhergestellt werden. Informationen zum Erstellen von Sicherungskopien für ZFS-Schnappschüsse finden Sie unter „Senden und Empfangen von ZFS-Daten“ auf Seite 227.

- „Erstellen und Löschen von ZFS-Schnappschüssen“ auf Seite 218
- „Anzeigen von und Zugreifen auf ZFS-Snapshots“ auf Seite 221
- „Wiederherstellen eines früheren ZFS-Snapshots“ auf Seite 223

Erstellen und Löschen von ZFS-Schnappschüssen

Schnappschüsse werden mithilfe des Befehls `zfs snapshot` erstellt. Diesem wird als einziges Argument der Name des zu erstellenden Schnappschusses übergeben. Der Schnappschussname ist wie folgt anzugeben:

filesystem@snapname
volume@snapname

Der Schnappschussname muss den unter „Konventionen für das Benennen von ZFS-Komponenten“ auf Seite 31 aufgeführten Benennungskonventionen genügen.

Im folgenden Beispiel wird ein Schnappschuss des Dateisystems `tank/home/cindy` mit dem Namen `friday` erstellt.

```
# zfs snapshot tank/home/cindy@friday
```

Mithilfe der Option `-r` können Sie Schnappschüsse für alle untergeordneten Dateisysteme erstellen. Beispiel:

```
# zfs snapshot -r tank/home@snap1
# zfs list -t snapshot -r tank/home
NAME                   USED  AVAIL  REFER  MOUNTPOINT
tank/home@snap1           0      -  2.11G  -
tank/home/cindy@snap1     0      -   115M  -
tank/home/lori@snap1      0      -  2.00G  -
tank/home/mark@snap1      0      -  2.00G  -
tank/home/tim@snap1       0      -  57.3M  -
```

Schnappschüsse besitzen keine konfigurierbaren Eigenschaften, und es können für Snapshots auch keine Dataset-Eigenschaften gesetzt werden. Beispiel:

```
# zfs set compression=on tank/home/cindy@friday
cannot set property for 'tank/home/cindy@friday':
this property can not be modified for snapshots
```

Schnappschüsse werden mithilfe des Befehls `zfs destroy` gelöscht. Beispiel:

```
# zfs destroy tank/home/cindy@friday
```

Ein Dataset kann nicht gelöscht werden, wenn Schnappschüsse dieses Datasets vorhanden sind. Beispiel:

```
# zfs destroy tank/home/cindy
cannot destroy 'tank/home/cindy': filesystem has children
use '-r' to destroy the following datasets:
tank/home/cindy@tuesday
tank/home/cindy@wednesday
tank/home/cindy@thursday
```

Wenn von einem Schnappschuss Klone erstellt wurden, müssen diese zuerst gelöscht werden, bevor das Löschen des Schnappschusses möglich ist.

Weitere Informationen zum Unterbefehl `destroy` finden Sie unter „Löschen eines ZFS-Dateisystems“ auf Seite 179.

Aufbewahren von ZFS-Schnappschüssen

Wenn Sie verschiedene automatische Schnappschuss-Richtlinien verwenden und dadurch ältere Schnappschüsse durch `zfs receive` gelöscht werden, weil sie nicht mehr auf der Sendeseite vorhanden sind, können Sie die Schnappschuss-Aufbewahrungsfunktion verwenden.

Durch die *Aufbewahrung* eines Schnappschusses wird verhindert, dass er gelöscht wird. Außerdem ermöglicht diese Funktion das Löschen eines Snapshots mit Klonen in Abhängigkeit von der Entfernung des letzten Klons mithilfe des Befehls `zfs destroy -d`. Jeder Schnappschuss besitzt eine zugeordnete Benutzerreferenzzählung, die bei Null beginnt. Dieser Wert wird um 1 erhöht, wenn ein Schnappschuss gesperrt wird, und um 1 verringert, wenn eine Sperre aufgehoben wird.

In der Vorgängerversion von Oracle Solaris konnte ein Schnappschuss nur dann mithilfe des Befehls `zfs destroy` gelöscht werden, wenn er keine Klone hatte. In dieser Oracle Solaris-Version muss zudem der Wert der Benutzerreferenzzählung des Schnappschusses auf Null stehen.

Sie können einen Schnappschuss oder eine Gruppe von Schnappschüssen aufbewahren. Anhand der folgenden Syntax wird beispielsweise ein Aufbewahrungs-Tag, `keep`, für `tank/home/cindy/snap@1` gesetzt:

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

Sie können die Option `-r` verwenden, um die Schnappschüsse aller untergeordneten Dateisystem rekursiv aufzubewahren. Beispiel:

```
# zfs snapshot -r tank/home@now
# zfs hold -r keep tank/home@now
```

Mithilfe dieser Syntax wird eine einzelne Referenz, `keep`, zu einem Schnappschuss oder einer Gruppe von Schnappschüssen hinzugefügt. Jeder Schnappschuss besitzt einen eigenen Tag-Namespace. Die Aufbewahrungs-Tags innerhalb dieses Namespace müssen eindeutig sein. Wenn ein Schnappschuss aufbewahrt wird, kann er nicht mithilfe des Befehls `zfs destroy` gelöscht werden. Beispiel:

```
# zfs destroy tank/home/cindy@snap1
cannot destroy 'tank/home/cindy@snap1': dataset is busy
```

Wenn Sie einen aufbewahrten Schnappschuss löschen möchten, verwenden Sie die Option `-d`. Beispiel:

```
# zfs destroy -d tank/home/cindy@snap1
```

Verwenden Sie den Befehl `zfs holds`, um eine Liste der aufbewahrten Schnappschüsse anzuzeigen. Beispiel:

```
# zfs holds tank/home@now
NAME           TAG   TIMESTAMP
tank/home@now  keep  Fri Aug  3 15:15:53 2012

# zfs holds -r tank/home@now
NAME                 TAG   TIMESTAMP
tank/home/cindy@now  keep  Fri Aug  3 15:15:53 2012
tank/home/lori@now   keep  Fri Aug  3 15:15:53 2012
tank/home/mark@now   keep  Fri Aug  3 15:15:53 2012
tank/home/tim@now    keep  Fri Aug  3 15:15:53 2012
tank/home@now        keep  Fri Aug  3 15:15:53 2012
```

Sie können den Befehl `zfs release` verwenden, um einen aufbewahrten Schnappschuss oder eine Gruppe aufbewahrter Schnappschüsse freizugeben. Beispiel:

```
# zfs release -r keep tank/home@now
```

Ist ein Schnappschuss freigegeben, kann er mithilfe des Befehls `zfs destroy` gelöscht werden. Beispiel:

```
# zfs destroy -r tank/home@now
```

Zwei neue Eigenschaften liefern Informationen zur Aufbewahrung von Schnappschüssen.

- Die Eigenschaft `defer_destroy` ist aktiviert (`on`), wenn der Schnappschuss zur späteren Löschung mithilfe des Befehls `zfs destroy -d` vorgesehen ist. Anderenfalls ist die Eigenschaft deaktiviert (`off`).
- Die Eigenschaft `userrefs` dient zur Angabe der Anzahl der Aufbewahrungen des Schnappschusses und wird auch als *Benutzerreferenzzählung* bezeichnet.

Umbenennen von ZFS-Schnappschüssen

Sie können Schnappschüsse umbenennen. Allerdings müssen Schnappschüsse innerhalb des Pools und Datasets, in dem sie erstellt wurden, umbenannt werden. Beispiel:

```
# zfs rename tank/home/cindy@snap1 tank/home/cindy@today
```

Außerdem entspricht die folgende Kurzsyntax der obigen Syntax:

```
# zfs rename tank/home/cindy@snap1 today
```

Der folgende Vorgang zum Umbenennen eines Schnappschusses wird nicht unterstützt, da sich Ziel-Pool und -Dateisystem von dem Pool und Dateisystem unterscheiden, in denen der betreffende Schnappschuss erstellt wurde:

```
# zfs rename tank/home/cindy@today pool/home/cindy@saturday
cannot rename to 'pool/home/cindy@today': snapshots must be part of same
dataset
```

Sie können Schnappschüsse mithilfe des Befehls `zfs rename -r` rekursiv umbenennen. Beispiel:

```
# zfs list -t snapshot -r users/home
NAME                         USED  AVAIL  REFER  MOUNTPOINT
users/home@now              23.5K      -  35.5K  -
users/home@yesterday            0      -    38K  -
users/home/lori@yesterday       0      -  2.00G  -
users/home/mark@yesterday       0      -  1.00G  -
users/home/neil@yesterday       0      -  2.00G  -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -t snapshot -r users/home
NAME                         USED  AVAIL  REFER  MOUNTPOINT
users/home@now              23.5K      -  35.5K  -
users/home@2daysago             0      -    38K  -
users/home/lori@2daysago        0      -  2.00G  -
users/home/mark@2daysago        0      -  1.00G  -
users/home/neil@2daysago        0      -  2.00G  -
```

Anzeigen von und Zugreifen auf ZFS-Snapshots

Sie können das Anzeigen von Schnappschusslisten in der `zfs list`-Ausgabe durch Verwenden der Pool-Eigenschaft `listsnapshots` aktivieren oder deaktivieren. Diese Eigenschaft ist standardmäßig aktiviert.

Wenn Sie diese Eigenschaft deaktivieren, können Sie Schnappschuss-Informationen mit dem Befehl `zfs list -t snapshot` anzeigen. Oder aktivieren Sie die Pool-Eigenschaft `listsnapshots`. Beispiel:

```
# 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
```

Schnappschüsse befinden sich im Verzeichnis `.zfs/snapshot` des Stammverzeichnisses des Dateisystems. Wenn das Dateisystem `tank/home/cindy` beispielsweise in `/home/dindy` eingehängt ist, befinden sich die Daten des Schnappschusses `tank/home/cindy@thursday` im Verzeichnis `/home/cindy/.zfs/snapshot/thursday` .

```
# ls /tank/home/cindy/.zfs/snapshot
thursday   tuesday    wednesday
```

Snapshots können wie folgt angezeigt werden:

```
# zfs list -t snapshot -r tank/home
NAME                        USED  AVAIL  REFER  MOUNTPOINT
tank/home/cindy@tuesday      45K      -  2.11G  -
tank/home/cindy@wednesday    45K      -  2.11G  -
tank/home/cindy@thursday       0      -  2.17G  -
```

Snapshots, die für ein bestimmtes Dateisystem erstellt wurden, können wie folgt angezeigt werden:

```
# zfs list -r -t snapshot -o name,creation tank/home
NAME                        CREATION
tank/home/cindy@tuesday     Fri Aug  3 15:18 2012
tank/home/cindy@wednesday   Fri Aug  3 15:19 2012
tank/home/cindy@thursday    Fri Aug  3 15:19 2012
tank/home/lori@today        Fri Aug  3 15:24 2012
tank/home/mark@today        Fri Aug  3 15:24 2012
```

Berechnung von Festplattenkapazität für ZFS-Schnappschüsse

Nach der Erstellung eines Schnappschusses wird dessen Festplattenkapazität anfänglich vom Schnappschuss und vom Dateisystem (sowie eventuell von früheren Schnappschüssen) gemeinsam genutzt. Wenn sich ein Dateisystem mit der Zeit ändert, wird gemeinsam genutzte Festplattenkapazität dann nur noch vom Schnappschuss belegt und in die Berechnung des vom Schnappschuss belegten Speicherplatzes (Eigenschaft `used`) einbezogen. Darüber hinaus kann durch das Löschen von Schnappschüssen die Festplattenkapazität, die Schnappschüssen eindeutig zugewiesen ist (und deswegen in der Eigenschaft *used* angegeben ist und somit belegt wird), größer werden.

Der Wert der Speicherplatzeigenschaft `referenced` des Schnappschusses ist identisch mit dem Wert bei dem Dateisystem, als der Schnappschuss erstellt wurde.

Sie können zusätzliche Informationen darüber erhalten, wie die Werte der Eigenschaft `used` belegt werden. Neue schreibgeschützte Dateisystem-Eigenschaften beschreiben die Belegung von Festplattenkapazität für Klone, Dateisysteme und Volumes. Beispiel:

```
$ zfs list -o space -r rpool
NAME                   AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
rpool                  59.1G  7.84G       21K    109K              0      7.84G
rpool@snap1                -    21K         -       -              -          -
rpool/ROOT             59.1G  4.78G         0     31K              0      4.78G
rpool/ROOT@snap1           -      0         -       -              -          -
rpool/ROOT/zfsBE       59.1G  4.78G      15.6M   4.76G              0          0
rpool/ROOT/zfsBE@snap1     -  15.6M         -       -              -          -
rpool/dump             59.1G  1.00G       16K   1.00G              0          0
rpool/dump@snap1           -    16K         -       -              -          -
rpool/export           59.1G    99K       18K     32K              0         49K
rpool/export@snap1         -    18K         -       -              -          -
rpool/export/home      59.1G    49K       18K     31K              0          0
rpool/export/home@snap1    -    18K         -       -              -          -
rpool/swap             61.2G  2.06G         0     16K           2.06G          0
rpool/swap@snap1           -      0         -       -              -          -
```

Eine Beschreibung dieser Eigenschaften können Sie Tabelle 5–1 entnehmen.

Wiederherstellen eines früheren ZFS-Snapshots

Mit dem Befehl `zfs rollback` können Sie alle Änderungen rückgängig machen, die seit der Erstellung eines bestimmten Schnappschusses an einem Dateisystem vorgenommen wurden. Im Dateisystem wird der Zustand zum Zeitpunkt der Erstellung des betreffenden Schnappschusses wiederhergestellt. Standardmäßig stellt dieser Befehl stets den Zustand des zuletzt gemachten Schnappschusses wieder her.

Damit das Dateisystem im Zustand eines früheren Schnappschusses wiederhergestellt werden kann, müssen alle dazwischen liegenden Schnappschusses gelöscht werden. Frühere Schnappschüsse können mithilfe der Option `-r` gelöscht werden.

Wenn Klone dazwischen liegender Snapshots vorhanden sind, müssen auch diese Klone mithilfe der Option `-R` gelöscht werden.

Hinweis – Das Dateisystem, dessen früherer Zustand wiederhergestellt werden soll, wird aus- und wieder eingehängt, wenn es gerade eingehängt ist. Wenn das betreffende Dateisystem nicht ausgehängt werden kann, schlägt die Wiederherstellung des früheren Zustands fehl. Die Option `-f` erzwingt bei Bedarf das Aushängen des Dateisystems.

Im folgenden Beispiel wird das Dateisystem `tank/home/cindy` auf den Schnappschuss mit dem Namen `serviceag` zurückgesetzt:

```
# zfs rollback tank/home/cindy@tuesday
cannot rollback to 'tank/home/cindy@tuesday': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
tank/home/cindy@wednesday
tank/home/cindy@thursday
# zfs rollback -r tank/home/cindy@tuesday
```

In diesem Beispiel wurden die Schnappschüsse `wednesday` und `thursday` gelöscht, da Sie den Zustand des davor liegenden Schnappschusses `tuesday` wiederhergestellt haben.

```
# zfs list -r -t snapshot -o name,creation tank/home/cindy
NAME                     CREATION
tank/home/cindy@tuesday  Fri Aug  3 15:18 2012
```

Ermitteln von ZFS-Schnappschussunterschieden (`zfs diff`)

Sie können ZFS-Schnappschussunterschiede mithilfe des Befehls `zfs diff` ermitteln.

Beispielsweise werden folgende zwei Schnappschüsse erstellt:

```
$ ls /tank/home/tim
fileA
$ zfs snapshot tank/home/tim@snap1
$ ls /tank/home/tim
fileA  fileB
$ zfs snapshot tank/home/tim@snap2
```

Um beispielsweise die Unterschiede zwischen zwei Schnappschüssen zu ermitteln, verwenden Sie folgende Syntax:

```
$ zfs diff tank/home/tim@snap1 tank/home/tim@snap2
M       /tank/home/tim/
+       /tank/home/tim/fileB
```

In der Ausgabe gibt `M` an, dass das Verzeichnis geändert wurde. Das `+` gibt an, dass `fileB` im späteren Schnappschuss vorhanden ist.

Das `R` in der folgenden Ausgabe gibt an, dass eine Datei in einem Schnappschuss umbenannt wurde.

```
$ mv /tank/cindy/fileB /tank/cindy/fileC
$ zfs snapshot tank/cindy@snap2
$ zfs diff tank/cindy@snap1 tank/cindy@snap2
M       /tank/cindy/
R       /tank/cindy/fileB -> /tank/cindy/fileC
```

In der folgenden Tabelle werden die vom Befehl `zfs diff` ermittelten Datei- oder Verzeichnisänderungen zusammengefasst.

Datei- oder Verzeichnisänderung	Kennung
Datei oder Verzeichnis bzw. Verknüpfung von Datei oder Verzeichnis wurde geändert	`M`

Datei- oder Verzeichnisänderung	Kennung
Datei oder Verzeichnis ist im älteren Schnappschuss vorhanden, nicht aber im neueren	–
Datei oder Verzeichnis ist im neueren Schnappschuss vorhanden, nicht aber im älteren	+
Datei oder Verzeichnis wurde umbenannt	R

Weitere Informationen finden Sie in der Manpage `zfs(1M)`.

Überblick über ZFS-Klone

Ein *Klon* ist ein schreibbares Volume bzw. Dateisystem, dessen anfänglicher Inhalt gleich dem des Datasets ist, von dem er erstellt wurde. Ebenso wie Schnappschüssen werden auch Klone sehr schnell erstellt und belegen anfänglich keine zusätzliche Festplattenkapazität. Darüber hinaus können Sie Snapshots von Klonen erstellen.

Klone können nur von Schnappschüssen erstellt werden. Beim Klonen eines Schnappschusses wird zwischen dem Klon und dem Schnappschuss eine implizite Abhängigkeit erstellt. Obwohl der betreffende Klon an anderer Stelle der Dateisystemhierarchie erstellt wird, kann der ursprüngliche Schnappschuss nicht gelöscht werden, solange von ihm ein Klon vorhanden ist. Die Eigenschaft `origin` enthält diese Abhängigkeit, und mit dem Befehl `zfs destroy` können solche Abhängigkeiten aufgelistet werden, falls sie vorhanden sind.

Klone erben keine Eigenschaften von dem Dataset, von dem sie erstellt wurden. Mit den Befehlen `zfs get` und `zfs set` können Sie die Eigenschaften eines geklonten Datasets anzeigen und ändern. Weitere Informationen zum Setzen von Eigenschaften von ZFS-Datasets finden Sie unter „Setzen von ZFS-Eigenschaften“ auf Seite 198.

Da ein Klon seine gesamte Festplattenkapazität anfänglich mit dem ursprünglichen Schnappschuss gemeinsam nutzt, ist sein Eigenschaftswert `used` zu Beginn auf null gesetzt. Wenn am Klon Änderungen vorgenommen werden, belegt er dementsprechend mehr Festplattenkapazität. Die Eigenschaft `used` des ursprünglichen Schnappschusses berücksichtigt keinen vom Klon belegten Speicherplatz.

- „Erstellen eines ZFS-Klons“ auf Seite 225
- „Löschen eines ZFS-Klons“ auf Seite 226
- „Ersetzen eines ZFS-Dateisystems durch einen ZFS-Klon“ auf Seite 226

Erstellen eines ZFS-Klons

Klone werden mit dem Befehl `zfs clone` erstellt. Sie müssen den Schnappschuss, von dem der Klon erstellt werden soll, sowie den Namen des neuen Dateisystems bzw. Volumes angeben. Das neue Dateisystem bzw. Volume kann sich an beliebiger Stelle innerhalb der ZFS-Hierarchie

befinden. Der Typ (z. B. Dateisystem oder Volume) des neuen Datasets entspricht dem des Schnappschusses, aus dem der Klon erstellt wurde. Sie können von einem Dateisystem in einem anderen Pool als dem des Schnappschusses des ursprünglichen Dateisystems keinen Klon erstellen.

Im folgenden Beispiel wird ein neuer Klon `tank/home/matt/bug123` mit dem gleichen anfänglichen Inhalt wie der des Schnappschusses `tank/ws/gate@yesterday` erstellt:

```
# zfs snapshot tank/ws/gate@yesterday
# zfs clone tank/ws/gate@yesterday tank/home/matt/bug123
```

Im folgenden Beispiel wird für einen temporären Benutzer aus dem Schnappschuss `projects/newproject@today` ein geklonter Arbeitsbereich namens `projects/teamA/tempuser` erstellt. Danach werden die Eigenschaften des geklonten Arbeitsbereichs gesetzt.

```
# 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
```

Löschen eines ZFS-Klons

ZFS-Klone werden mithilfe des Befehls `zfs destroy` gelöscht. Beispiel:

```
# zfs destroy tank/home/matt/bug123
```

Bevor ein übergeordneter Snapshot gelöscht werden kann, müssen zunächst seine Klone gelöscht werden.

Ersetzen eines ZFS-Dateisystems durch einen ZFS-Klon

Mit dem Befehl `zfs promote` können Sie ein aktives ZFS-Dateisystem durch einen Klon dieses Dateisystems ersetzen. Diese Funktion ermöglicht das Klonen und Ersetzen von Dateisystemen, sodass das *ursprüngliche* Dateisystem der Klon des betreffenden Dateisystems werden kann. Darüber hinaus ermöglicht diese Funktion das Löschen des Dateisystems, von dem der Klon ursprünglich erstellt wurde. Ohne diese "Klon-Promotion" können ursprüngliche Dateisysteme, in denen aktive Klone enthalten sind, nicht gelöscht werden. Weitere Informationen zum Löschen von Klonen finden Sie unter „Löschen eines ZFS-Klons“ auf Seite 226.

Im folgenden Beispiel wird das Dateisystem `tank/test/produktA` geklont. Anschließend wird das geklonte Dateisystem (`tank/test/produktAbeta`) zum ursprünglichen Dateisystem `tank/test/produktA` gemacht.

```
# 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  -
```

In dieser Ausgabe des Befehls `zfs list` sehen Sie, dass die Festplattenkapazitätsangabe des ursprünglichen Dateisystems `produktA` durch die des Dateisystems `produktAbeta` ersetzt wurde.

Sie können den Ersetzungsvorgang durch Umbenennen der Dateisysteme abschließen. Beispiel:

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
```

Optional können Sie auch das alte Dateisystem entfernen. Beispiel:

```
# zfs destroy tank/test/productAlegacy
```

Senden und Empfangen von ZFS-Daten

Der Befehl `zfs send` erstellt von Schnappschüssen Datenstrominstanzen, die auf die Standardausgabe geschrieben werden. Standardmäßig wird ein vollständiger Datenstrom erzeugt. Sie können die Ausgabe in eine Datei oder ein anderes Dateisystem umleiten. Der Befehl `zfs receive` erstellt einen Schnappschuss, dessen Inhalt in einem Datenstrom auf der Standardeingabe angegeben wird. Wenn ein vollständiger Datenstrom gelesen wurde, wird ein neues Dateisystem erstellt. Mit diesen Befehlen können Sie ZFS-Schnappschussdaten senden und ZFS-Schnappschussdaten und Dateisysteme empfangen. Siehe hierzu die Beispiele im nachfolgenden Abschnitt.

- „Sichern von ZFS-Daten mit anderen Softwarepaketen zur Erstellung von Sicherungskopien“ auf Seite 229
- „Senden von ZFS-Schnappschüssen“ auf Seite 231
- „Empfangen von ZFS-Schnappschüssen“ auf Seite 232
- „Anwenden verschiedener Eigenschaftswerte auf einen ZFS-Schnappschussdatenstrom“ auf Seite 234

- „Senden und Empfangen komplexer ZFS-Snapshot-Datenstreams“ auf Seite 236
- „Replikation von ZFS-Daten über das Netzwerk“ auf Seite 239

Die folgenden Lösungen zum Sichern von ZFS-Daten stehen zur Verfügung:

- **Sicherungsprodukte für Unternehmen** – Wenn Sie Bedarf an folgenden Leistungsmerkmalen haben, sollten Sie eine Unternehmenslösung für die Datensicherung in Betracht ziehen:
  - Wiederherstellung auf Dateibasis
  - Überprüfung der Datensicherungsmedien
  - MedienAdministration
- **Dateisystem-Schnappschüsse und Wiederherstellen des früheren Zustands eines Dateisystems** – Mit den Befehlen `zfs snapshot` und `zfs rollback` können Sie auf einfache Weise Kopien von Dateisystemen erstellen und aus diesen bei Bedarf Dateisysteme auf einen früheren Zustand zurücksetzen. Sie können die Lösung beispielsweise verwenden, wenn Sie ein Dateisystem bzw. Dateien aus einer früheren Dateisystemversion wiederherstellen möchten.

  Weitere Informationen zum Erstellen von Schnappschüssen und Wiederherstellen einer früheren Version aus einem Schnappschuss finden Sie unter „Überblick über ZFS-Schnappschüsse“ auf Seite 217.
- Sichern von Schnappschüssen – Mit den Befehlen `zfs send` und `zfs receive` können Sie ZFS-Schnappschüsse senden und empfangen. Sie können inkrementelle Änderungen zwischen Schnappschüssen sichern, Dateien können jedoch nicht einzeln wiederhergestellt werden. Stattdessen muss der gesamte Dateisystem-Schnappschuss wiederhergestellt werden. Diese Befehle stellen keine vollständige Sicherungslösung für Ihre ZFS-Daten dar.
- **Replikation über Netzwerk** – Mit den Befehlen `zfs send` und `zfs receive` können Sie Dateisysteme von einem System auf ein anderes kopieren. Dieser Vorgang unterscheidet sich von herkömmlichen Praktiken bei Software zur DatenträgerAdministration, die Datenspeichergeräte über WAN spiegeln. Dafür ist keine spezielle Konfiguration bzw. Hardware erforderlich. Der Vorteil der Replikation eines ZFS-Dateisystems besteht darin, dass das betreffende Dateisystem in einem Speicher-Pool eines anderen Systems neu erstellt werden kann und Sie für diesen neuen Pool verschiedene Replikationsmethoden wie z. B. RAID-Z angeben können, die Dateisystemdaten aber gleich bleiben.
- Archivierungsserviceprogramme – Speichern von ZFS-Daten mit Archivierungsserviceprogrammen wie `tar`, `cpio` und `pax` oder Datensicherungssoftware von Drittherstellern. Derzeit konvertieren `tar` und `cpio` die NFSv4-basierten Zugriffskontrolllisten korrekt, `pax` hingegen nicht.

Sichern von ZFS-Daten mit anderen Softwarepaketen zur Erstellung von Sicherungskopien

Neben den Befehlen `zfs send` und `zfs receive` können Sie zum Sichern von ZFS-Dateien auch Archivierungsserviceprogramme wie z. B. `tar` und `cpio` verwenden. Diese Serviceprogramme sichern ZFS-Dateiattribute und -Zugriffskontrolllisten und können diese auch wiederherstellen. Überprüfen Sie die entsprechenden Optionen der Befehle `tar` und `cpio`.

Aktuelle Informationen zu Problemen im Zusammenhang mit ZFS und Backupsoftware von Drittherstellern finden Sie in den Solaris 10-Versionshinweisen.

Identifizieren von ZFS-Schnappschuss-Streams

Ein Schnappschuss eines ZFS-Dateisystems oder -Volumes wird mit dem Befehl `zfs send` in einen Schnappschuss-Stream konvertiert. Danach können Sie den Schnappschuss-Stream verwenden, um ein ZFS-Dateisystem oder -Volume mit dem Befehl `zfs receive` neu zu erstellen.

Je nach den `zfs send`-Optionen, die für das Erstellen des Schnappschuss-Streams verwendet wurden, werden unterschiedliche Typen von Stream-Formaten generiert.

- Vollständiger Stream – Besteht aus dem gesamten Dataset-Inhalt seit dem Erstellen des Datasets bis zu dem angegebenen Schnappschuss.

  Der Standard-Stream, der mit dem Befehl `zfs send` generiert wird, ist ein vollständiger Stream. Er enthält ein Dateisystem oder Volume bis zu dem angegebenen Schnappschuss einschließlich. Dieser Stream enthält keine anderen Schnappschüsse als den in der Befehlszeile angegebenen Schnappschuss.
- Inkrementeller Stream - Besteht aus der Differenz zwischen zwei Schnappschüssen.

Ein Stream-Package ist ein Stream-Typ, der mindestens einen vollständigen oder inkrementellen Stream enthält. Es gibt drei Typen von Stream-Packages:

- Replikations-Stream-Package – Besteht aus dem angegebenen Dataset und seinen untergeordneten Datasets. Es umfasst alle Zwischenschnappschüsse. Wenn der Ursprung des geklonten Datasets kein untergeordneter Schnappschuss des Schnappschusses ist, der in der Befehlszeile angegeben wird, wird dieses ursprüngliche Dataset nicht in dem Stream-Package aufgenommen. Zum Empfang des Streams muss das ursprüngliche Dataset im Zielspeicherpool vorhanden sein.

  Beachten Sie die folgende Liste der Datasets und ihrer Ursprünge: Angenommen, sie wurden in der Reihenfolge erstellt, in der sie unten angezeigt werden.

  ```
  NAME                    ORIGIN
  pool/a                  -
  pool/a/1                -
  ```

```
pool/a/1@clone          -
pool/b                  -
pool/b/1                pool/a/1@clone
pool/b/1@clone2         -
pool/b/2                pool/b/1@clone2
pool/b@pre-send         -
pool/b/1@pre-send       -
pool/b/2@pre-send       -
pool/b@send             -
pool/b/1@send           -
pool/b/2@send           -
```

Ein Replikations-Stream-Package, das mit der folgenden Syntax erstellt wird:

```
# zfs send -R pool/b@send ....
```

besteht aus den folgenden vollständigen und inkrementellen Streams:

```
TYPE    SNAPSHOT                INCREMENTAL FROM
full    pool/b@pre-send         -
incr    pool/b@send             pool/b@pre-send
incr    pool/b/1@clone2         pool/a/1@clone
incr    pool/b/1@pre-send       pool/b/1@clone2
incr    pool/b/1@send           pool/b/1@send
incr    pool/b/2@pre-send       pool/b/1@clone2
incr    pool/b/2@send           pool/b/2@pre-send
```

In der vorherigen Ausgabe ist der Schnappschuss `pool/a/1@clone` nicht in dem Replikations-Stream-Package enthalten. Als solches kann dieses Replikations-Stream-Package nur in einem Pool empfangen werden, der bereits den Schnappschuss `pool/a/1@clone` enthält.

- Rekursives Stream-Package – Besteht aus dem angegebenen Dataset und seinen untergeordneten Datasets. Im Gegensatz zu Replikations-Stream-Packages sind keine Zwischenschnappschüsse enthalten, es sei denn, sie sind der Ursprung eines geklonten Datasets, das in dem Stream enthalten ist. Wenn der Ursprung eines Datasets kein untergeordneter Schnappschuss des Schnappschusses ist, der in der Befehlszeile angegeben wird, entspricht das Behavior standardmäßig dem Behavior von Replikations-Streams. Ein eigenständiger rekursiver Stream, wie unten beschrieben, wird jedoch so erstellt, dass keine externen Abhängigkeiten vorhanden sind.

  Ein rekursives Stream-Package, das mit der folgenden Syntax erstellt wird:

  ```
  # zfs send -r pool/b@send ...
  ```

  besteht aus den folgenden vollständigen und inkrementellen Streams:

  ```
  TYPE    SNAPSHOT                INCREMENTAL FROM
  full    pool/b@send             -
  incr    pool/b/1@clone2         pool/a/1@clone
  incr    pool/b/1@send           pool/b/1@clone2
  incr    pool/b/2@send           pool/b/1@clone2
  ```

  In der vorherigen Ausgabe ist der Schnappschuss `pool/a/1@clone` nicht in dem rekursiven Stream-Package enthalten. Als solches kann dieses rekursive Stream-Package nur in einem

Pool empfangen werden, der bereits den Schnappschuss `pool/a/1@clone` enthält. Dieses Behavior entspricht dem Szenario des Replikations-Stream-Packages, das oben beschrieben wurde.

- Eigenständiges rekursives Stream-Package - Ist nicht von Datasets abhängig, die nicht in dem Stream-Package enthalten sind. Dieses rekursive Stream-Package wird mit der folgenden Syntax erstellt:

  ```
  # zfs send -rc pool/b@send ...
  ```

  Es besteht aus den folgenden vollständigen und inkrementellen Streams:

  ```
  TYPE    SNAPSHOT                INCREMENTAL FROM
  full    pool/b@send             -
  full    pool/b/1@clone2
  incr    pool/b/1@send           pool/b/1@clone2
  incr    pool/b/2@send           pool/b/1@clone2
  ```

  Beachten Sie, dass der eigenständige rekursive Stream einen vollständigen Stream des Schnappschusses `pool/b/1@clone2` enthält, sodass der Schnappschuss `pool/b/1` ohne externe Abhängigkeiten empfangen werden kann.

Senden von ZFS-Schnappschüssen

Der Befehl `zfs send` dient zum Senden der Kopie eines Schnappschuss-Datenstroms und zum Empfangen des Schnappschuss-Datenstroms in einem anderen Pool auf demselben System oder in einem anderen Pool auf einem anderen System, das zur Aufbewahrung von Sicherungsdaten verwendet wird. Beispiel: Zum Senden des Schnappschuss-Streams an einen anderen Pool auf demselben System verwenden Sie folgende Syntax:

```
# zfs send tank/dana@snap1 | zfs recv spool/ds01
```

Sie können `zfs recv` als Aliasnamen für den Befehl `zfs receive` verwenden.

Wenn Sie den Schnappschuss-Datenstrom an ein anderes System senden, setzen Sie für die Ausgabe von `zfs send` mit dem Befehl `ssh` eine Pipeline. Beispiel:

```
sys1# zfs send tank/dana@snap1 | ssh sys2 zfs recv newtank/dana
```

Wenn Sie einen vollständige Datenstrom senden, darf das Zieldateisystem nicht vorhanden sein.

Sie können inkrementelle Daten mit der Option `i` des Befehls `-zfs send` senden. Beispiel:

```
sys1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

Das erste Argument (`snap1`) ist der frühere und das zweite Argument (`snap2`) der spätere Schnappschuss. In diesem Fall muss das Dateisystem `newtank/dana` bereits vorhanden sein, damit die inkrementellen Daten empfangen werden können.

Hinweis – Wenn auf Dateiinformationen im ursprünglich empfangenen Dateisystem zugegriffen wird, kann dies dazu führen, dass der inkrementelle Schnappschussempfangsvorgang mit einer Meldung wie der Folgenden nicht erfolgreich verläuft:

```
cannot receive incremental stream of tank/dana@snap2 into newtank/dana:
most recent snapshot of tank/dana@snap2 does not match incremental source
```

Sie sollten die Eigenschaft `atime` auf "off" setzen, wenn Sie auf Dateiinformationen im ursprünglich empfangenen Dateisystem zugreifen müssen und wenn Sie gleichzeitig inkrementelle Schnappschüsse in dem empfangenen Dateisystem empfangen müssen.

Die inkrementelle Quelle *snap1* kann als letzte Komponente des Schnappschussnamens angegeben werden. Dies bedeutet, dass Sie nach dem Zeichen `@` für *snap1* nur den Namen angeben müssen, von dem angenommen wird, dass er zum gleichen System wie *snap2* gehört. Beispiel:

```
sys1# zfs send -i snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

Diese Kurzsyntax entspricht der im vorherigen Beispiel demonstrierten inkrementellen Syntax.

Die folgende Meldung wird angezeigt, wenn Sie versuchen, aus einem anderen Dateisystem (`snapshot1`) einen inkrementellen Datenstrom zu erzeugen.

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Wenn mehrere Kopien gespeichert werden sollen, kann eine Komprimierung der Datenstrominstanz des ZFS-Schnappschusses mithilfe des Befehls `gzip` nützlich sein. Beispiel:

```
# zfs send pool/fs@snap | gzip > backupfile.gz
```

Empfangen von ZFS-Schnappschüssen

Beim Empfangen von Datensystem-Snapshots sollten Sie folgende Schlüsselaspekte berücksichtigen:

- Sowohl der Schnappschuss als auch das Dateisystem werden empfangen.
- Das Dateisystem und alle untergeordneten Dateisysteme werden ausgehängt.
- Während des Empfangs kann auf die betreffenden Dateisysteme nicht zugegriffen werden.
- Das ursprüngliche Dateisystem, das empfangen werden soll, darf bei der Übertragung nicht vorhanden sein.
- Wenn der Dateisystemname bereits vorhanden ist, können Sie das Dateisystem mit dem Befehl `zfs rename` umbenennen.

Beispiel:

```
# 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
```

Wenn Sie am Zieldateisystem eine Änderung vornehmen und danach einen weiteren inkrementellen Schnappschuss senden möchten, müssen Sie zunächst den vorherigen Zustand des Zieldateisystems wiederherstellen.

Betrachten Sie das folgende Beispiel. Zunächst ändern Sie das Dateisystem wie folgt:

```
sys2# rm newtank/dana/file.1
```

Dann senden Sie einen weiteren inkrementellen Schnappschuss (`tank/dana@snap3`). Sie müssen jedoch erst den vorherigen Zustand des Zieldateisystems wiederherstellen, damit es den neuen inkrementellen Schnappschuss empfangen kann. Sie können den Wiederherstellungsschritt aber auch mithilfe der Option `-F` überspringen. Beispiel:

```
sys1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh sys2 zfs recv -F newtank/dana
```

Beim Empfang eines inkrementellen Snapshots muss das Zieldateisystem bereits vorhanden sein.

Wenn Sie am Dateisystem Änderungen vornehmen und den vorherigen Zustand des Zieldateisystems nicht wiederherstellen, sodass es den neuen inkrementellen Schnappschusses empfangen kann, oder Sie die Option `-F` nicht verwenden, wird eine Meldung wie die folgende angezeigt:

```
sys1# zfs send -i tank/dana@snap4 tank/dana@snap5 | ssh sys2 zfs recv newtank/dana
cannot receive: destination has been modified since most recent snapshot
```

Bevor das Ausführen der Option `-F` als erfolgreich gemeldet wird, werden die folgenden Überprüfungen durchgeführt:

- Wenn der letzte Schnappschuss nicht mit der inkrementellen Quelle identisch ist, wird weder die Wiederherstellung des früheren Zustands noch der Empfang abgeschlossen, und es wird eine Fehlermeldung angezeigt.
- Wenn Sie versehentlich den Namen eines anderen Dateisystems angeben, der mit dem inkrementellen Quellparameter des Befehls `zfs receive` nicht übereinstimmt, wird weder die Wiederherstellung des früheren Zustands noch der Empfang abgeschlossen, und es wird die folgende Fehlermeldung angezeigt:

  ```
  cannot send 'pool/fs@name': not an earlier snapshot from the same fs
  ```

Anwenden verschiedener Eigenschaftswerte auf einen ZFS-Schnappschussdatenstrom

Sie können einen ZFS-Schnappschussdatenstrom mit einem bestimmten Dateisystem-Eigenschaftswert senden, aber auch einen anderen lokalen Eigenschaftswert angeben, wenn der Schnappschussdatenstrom empfangen wird. Alternativ können Sie ebenfalls festlegen, dass der ursprüngliche Eigenschaftswert beim Empfang des Schnappschussdatenstroms verwendet wird, um das ursprüngliche Dateisystem neu zu erstellen. Außerdem besteht die Möglichkeit, eine Dateisystemeigenschaft zu deaktivieren, wenn der Schnappschussdatenstrom empfangen wird.

- Verwenden Sie `zfs inherit -S`, um einen lokalen Eigenschaftswert auf den empfangenen Wert (sofern vorhanden) zurückzusetzen. Wenn eine Eigenschaft keinen empfangenen Wert aufweist, ist das Verhalten des Befehls `zfs inherit -S` dasselbe wie das des Befehls `zfs inherit` ohne die Option `-S`. Wenn die Eigenschaft einen empfangenen Wert aufweist, maskiert der Befehl `zfs inherit` den empfangenen Wert mit dem vererbten Wert, bis dieser durch Ausgabe des Befehls `zfs inherit -S` auf den empfangenen Wert zurückgesetzt wird.
- Sie können den Befehl `zfs get -o` verwenden, um die neue nicht standardmäßige Spalte `RECEIVED` einzubeziehen. Sie können aber auch den Befehl `zfs get -o all` verwenden, um alle Spalten einschließlich `RECEIVED` einzubeziehen.
- Sie können die Option `zfs send -p` verwenden, um Eigenschaften in zu sendenden Datenstrom ohne die Option `-R` einzubeziehen.
- Mit dem Befehl `zfs receive -e` können mit dem letzten Element des gesendeten Schnappschussnamens den Namen des neuen Schnappschusses bestimmen. Im folgenden Beispiel wird der Schnappschuss `poola/bee/cee@1` an das Dateisystem `poold/eee` gesendet, und nur das letzte Element (`cee@1`) des Schnappschussnamens wird verwendet, um das Dateisystem und den Schnappschuss zu erstellen.

```
# 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  -
```

In einigen Fällen kann es vorkommen, dass Dateisystemeigenschaften in einem gesendeten Datenstrom nicht für das empfangene Dateisystem übernommen werden, oder lokale Dateisystemeigenschaften, wie z. B. der Eigenschaftswert `mountpoint`, zu Problemen mit einer Wiederherstellung führen.

Für das Dateisystem `tank/data` ist beispielsweise die Eigenschaft `compression` deaktiviert. Ein Schnappschuss des Dateisystems `tank/data` wird mit Eigenschaften (Option `-p`) an einen Sicherungs-Pool gesendet und mit aktivierter Eigenschaft `compression` empfangen.

```
# zfs get compression tank/data
NAME       PROPERTY     VALUE     SOURCE
tank/data  compression  off       default
# zfs snapshot tank/data@snap1
# zfs send -p tank/data@snap1 | zfs recv -o compression=on -d bpool
# zfs get -o all compression bpool/data
NAME        PROPERTY     VALUE     RECEIVED  SOURCE
bpool/data  compression  on        off       local
```

In diesem Beispiel wird die Eigenschaft `compression` aktiviert, wenn der Schnappschuss in `bpool` empfangen wird. Demzufolge ist der Wert von `compression` für `bpool/data` aktiviert.

Wird dieser Schnappschussdatenstrom an einen neuen Pool (`restorepool`) gesendet, möchten Sie eventuell aus Gründen der Wiederherstellung alle ursprünglichen Schnappschusseigenschaften beibehalten. In diesem Fall müssten Sie den Befehl `zfs send -b` verwenden, um die ursprünglichen Schnappschusseigenschaften wiederherzustellen. Beispiel:

```
# zfs send -b bpool/data@snap1 | zfs recv -d restorepool
# zfs get -o all compression restorepool/data
NAME              PROPERTY     VALUE     RECEIVED  SOURCE
restorepool/data  compression  off       off       received
```

In diesem Beispiel ist der Komprimierungswert `off`. Dieser stellt den Wert für die Schnappschusskomprimierung aus dem ursprünglichen `tank/data`-Dateisystem dar.

Wenn Sie bei einem lokalen Dateisystem-Eigenschaftswert in einem Schnappschussdatenstrom möchten, dass die Eigenschaft beim Empfang deaktiviert ist, verwenden Sie den Befehl `zfs receive -x`. Der folgende Befehl sendet beispielsweise einen rekursiven Schnappschussdatenstrom von Dateisystemen für Home-Verzeichnisse mit allen für einen Sicherungs-Pool reservierten Dateisystemeigenschaften, aber ohne die Eigenschaftswerte für das Kontingent:

```
# zfs send -R tank/home@snap1 | zfs recv -x quota bpool/home
# zfs get -r quota bpool/home
NAME                   PROPERTY  VALUE  SOURCE
bpool/home             quota     none   local
bpool/home@snap1       quota     -      -
bpool/home/lori        quota     none   default
bpool/home/lori@snap1  quota     -      -
bpool/home/mark        quota     none   default
bpool/home/mark@snap1  quota     -      -
```

Wenn der rekursive Schnappschuss nicht mit der Option `-x` empfangen wurde, werden die Kontingenteigenschaften in den empfangenen Dateisystemen festgelegt.

```
# zfs send -R tank/home@snap1 | zfs recv bpool/home
# zfs get -r quota bpool/home
NAME                   PROPERTY  VALUE  SOURCE
```

```
bpool/home              quota     none   received
bpool/home@snap1        quota     -      -
bpool/home/lori         quota     10G    received
bpool/home/lori@snap1   quota     -      -
bpool/home/mark         quota     10G    received
bpool/home/mark@snap1   quota     -      -
```

Senden und Empfangen komplexer ZFS-Snapshot-Datenstreams

In diesem Abschnitt wird das Senden und Empfangen komplexerer Snapshot-Datenstreams mit den Befehlen `zfs send -I` und `-R` beschrieben.

Beachten Sie beim Senden und Empfangen von komplexen ZFS-Schnappschuss-Datenströmen Folgendes:

- Verwenden Sie `zfs send` mit der Option `-I`, um alle inkrementellen Datenströme aus einem Schnappschuss an einen kumulativen Schnappschuss zu senden. Sie können mithilfe dieser Option aber auch einen inkrementellen Datenstrom aus dem ursprünglichen Schnappschuss senden, um einen Klon zu erstellen. Der ursprüngliche Schnappschuss muss auf der Empfangsseite bereits vorhanden sein, damit der inkrementelle Datenstrom angenommen werden kann.
- Mit `zfs send` und der Option `-R` senden Sie einen Replikationsdatenstrom aller untergeordneten Dateisysteme. Nach dem Empfang des Replikationsdatenstroms werden alle Eigenschaften, Schnappschüsse, abhängigen Dateisysteme und Klone beibehalten.
- Wenn die Option `zfs send -r` ohne die Option `-c` verwendet wird und wenn die Option `zfs send -R` verwendet wird, lassen Stream-Packages den `Ursprung` von Klons unter bestimmten Umständen weg. Weitere Informationen finden Sie in „Identifizieren von ZFS-Schnappschuss-Streams“ auf Seite 229.
- Verwenden Sie beide Optionen, um einen inkrementellen Replikationsdatenstrom zu senden.
  - Änderungen an Eigenschaften werden beibehalten, ebenso wie Namensänderungen von Schnappschüssen und Dateisystemen und Löschvorgänge.
  - Wenn `zfs recv -F` beim Empfang des Replikationsdatenstroms nicht angegeben ist, werden Löschvorgänge von Datasets ignoriert. In diesem Fall behält die Syntax `zfs recv -F` auch die Bedeutung *Bei Bedarf Rollback* bei.
  - Wie in anderen Fällen (außer `zfs send -R`) mit `-i` oder `-I` werden bei Verwendung von `-I` alle Schnappschüsse zwischen snapA und snapD gesendet. Bei Verwendung von `-i` wird nur `snapD` (für sämtliche untergeordneten Objekte) gesendet.
- Der Empfang dieser mithilfe des Befehls `zfs send` gesendeten neuen Datenströme setzt voraus, dass auf dem empfangenden System eine Softwareversion ausgeführt wird, die in der Lage ist, diese Datenströme zu senden. Die Version des Datenstroms wird inkrementiert.

Es ist jedoch möglich, auf Datenströme aus älteren Pool-Versionen über neuere Softwareversionen zuzugreifen. So können Sie beispielsweise Datenströme, die mit den neueren Optionen erstellt wurden, an und aus Pools der Version 3 senden. Zum Empfangen eines mit den neueren Optionen gesendeten Datenstroms muss jedoch aktuelle Software ausgeführt werden.

BEISPIEL 6–1 Senden und Empfangen komplexer ZFS-Schnappschuss-Datenströme

Eine Gruppe inkrementeller Schnappschüsse lässt sich mithilfe von `zfs send` und der Option `-I` zu einem Schnappschuss kombinieren. Beispiel:

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

Anschließend entfernen Sie `snapB`, `snapC` und `snapD`.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Um den kombinierten Schnappschuss zu empfangen, verwenden Sie den folgenden Befehl.

```
# 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  -
```

Außerdem können Sie mit `zfs send -I` einen Schnappschuss und einen Klon-Schnappschuss zu einem kombinierten Dataset verbinden. Beispiel:

```
# 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
```

Mit dem Befehl `zfs send -R` können Sie ein ZFS-Dateisystem und alle untergeordneten Dateisysteme bis hin zum benannten Schnappschuss replizieren. Nach dem Empfang dieses Datenstroms werden alle Eigenschaften, Schnappschüsse, abhängigen Dateisysteme und Klone beibehalten.

Im folgenden Beispiel werden Schnappschüsse für Benutzerdateisysteme erstellt. Es wird ein Replikationsdatenstrom für alle Benutzer-Schnappschüsse erstellt. Anschließend werden die ursprünglichen Dateisysteme und Schnappschüsse gelöscht und wiederhergestellt.

BEISPIEL 6–1 Senden und Empfangen komplexer ZFS-Schnappschuss-Datenströme *(Fortsetzung)*

```
# 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  -
```

Im folgenden Beispiel wird der Befehl `zfs send -R` verwendet, um das Dateisystem `users` und seine untergeordneten Objekte zu replizieren und den replizierten Datenstrom an einen anderen Pool, `users2`, zu senden.

```
# 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
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  -
```

Replikation von ZFS-Daten über das Netzwerk

Mit den Befehlen `zfs send` und `zfs recv` können Sie eine Datenstrominstanz über das Netzwerk von einem System auf ein anderes kopieren. Beispiel:

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Mithilfe des Befehls wird der Schnappschuss `tank/cindy@today` gesendet und vom Dateisystem `sandbox/restfs` empfangen. Außerdem wird ein `restfs@today`-Schnappschuss für das System `newsys` erstellt. In diesem Beispiel wird auf dem entfernten System `ssh` verwendet.

◆ ◆ ◆ KAPITEL 7

Schützen von Oracle Solaris ZFS-Dateien mit Zugriffskontrolllisten und Attributen

Dieses Kapitel enthält Informationen zum Arbeiten mit Zugriffssteuerungslisten. Solche Listen schützen ZFS-Dateien, weil sie im Vergleich zu den UNIX-Standardzugriffsrechten feiner abgestimmte Zugriffsrechte definieren und anwenden.

Dieses Kapitel enthält die folgenden Abschnitte:

Solaris-Modell zu Zugriffskontrolllisten

Frühere Versionen des Betriebssystems Solaris unterstützten eine auf der POSIX-Spezifikation beruhende Implementierung von Zugriffskontrolllisten. POSIX-basierte Zugriffskontrolllisten dienen zum Schutz von UFS-Dateien und werden von NFS-Versionen vor NFSv4 verwendet.

Seit der Einführung von NFSv4 unterstützt ein neues Modell für Zugriffskontrolllisten vollständig die Interoperabilität, die NFSv4 für die Kommunikation zwischen UNIX-Clients und anderen Clients bietet. Die neue, in der NFSv4-Spezifikation definierte Implementierung von Zugriffskontrolllisten bietet eine reichhaltigere Semantik, die auf NT-basierten Zugriffskontrolllisten beruht.

Die Hauptunterschiede des neuen Zugriffskontrolllistenmodells bestehen in Folgendem:

- Basiert auf der NFSv4-Spezifikation und ähnelt NT-Zugriffskontrolllistenmodellen,
- enthält einen feiner abstimmbaren Satz an Zugriffsrechten, Weitere Informationen finden Sie in Tabelle 7–2.

- wird mit den Befehlen `chmod` und `ls` anstatt `setfacl` und `getfacl` eingestellt und angezeigt,
- enthält eine reichhaltigere Vererbungssemantik, um festlegen zu können, wie Zugriffsrechte von über- und untergeordneten Verzeichnissen geregelt werden, usw. Weitere Informationen dazu finden Sie in „Vererbung von Zugriffskontrolllisten“ auf Seite 247.

Beide Zugriffskontrolllistenmodelle bieten eine feiner abstimmbare Kontrolle von Zugriffsrechten als die Standardzugriffsrechte. Wie bei POSIX-basierten Zugriffskontrolllisten bestehen auch die neuen Zugriffskontrolllisten aus mehreren Zugriffskontrolleinträgen.

POSIX-basierte Zugriffskontrolllisten definieren mithilfe eines einzigen Eintrags, welche Zugriffsrechte zulässig und unzulässig sind. Das neue Zugriffskontrolllistenmodell besitzt zwei Arten von Zugriffskontrolleinträgen, die sich auf die Überprüfung von Zugriffsrechten auswirken: `ALLOW` (Erlauben) und `DENY` (Verweigern). Demzufolge können Sie nicht aus einem einzigen Zugriffskontrolleintrag schließen, ob die in diesem Zugriffskontrolleintrag nicht definierten Zugriffsrechte zulässig sind oder nicht.

Die Konvertierung zwischen NFSv4-basierten und POSIX-basierten Zugriffskontrolllisten geschieht wie folgt:

- Bei der Verwendung von Serviceprogrammen, die Zugriffsrechte überprüfen (z. B. die Befehle `cp`, `mv`, `tar`, `cpio` oder `rcp`) werden die POSIX-basierten Zugriffskontrolllisten in die entsprechenden NFSv4-basierten Zugriffskontrolllisten konvertiert, damit UFS-Dateien mit Zugriffskontrolllisten in ein ZFS-Dateisystem transferiert werden können.
- Einige NFSv4-basierte Zugriffskontrolllisten werden in POSIX-basierte Zugriffskontrolllisten konvertiert. Wenn eine NFSv4–basierte Zugriffskontrollliste nicht in eine POSIX-basierte Zugriffskontrollliste konvertiert werden kann, wird in etwas die folgende Meldung angezeigt:

  ```
  # cp -p filea /var/tmp
  cp: failed to set acl entries on /var/tmp/filea
  ```
- Wenn Sie auf einem System, auf dem die neueste Solaris-Version installiert ist, mit `tar` oder `cpio` ein UFS-Archiv mit der Option zur Beibehaltung der Zugriffskontrollliste (`tar -p` bzw. `cpio -P`) erstellen, gehen beim Extrahieren des Archivs auf einem System, auf dem eine frühere Solaris-Version installiert ist, die Zugriffskontrolllisten verloren.

  Alle Dateien werden mit den ordnungsgemäßen Dateimodi extrahiert, aber die Zugriffskontrolllisten werden ignoriert.
- Der Befehl `ufsrestore` ermöglicht es, Daten in einem ZFS-Dateisystem wiederherzustellen. Wenn die Originaldaten POSIX-basierte Zugriffskontrolllisten enthalten, werden diese in NFSv4-basierte Zugriffskontrolllisten konvertiert.
- Wenn Sie versuchen, eine NFSv4-basierte Zugriffskontrollliste an einer UFS-Datei zu setzen, wird eine Meldung wie die folgende angezeigt:

  ```
  chmod: ERROR: ACL type's are different
  ```

- Wenn Sie versuchen, eine POSIX-basierte Zugriffskontrollliste an einer ZFS-Datei zu setzen, wird eine Meldung wie die folgende angezeigt:

```
# getfacl filea
File system doesn't support aclent_t style ACL's.
See acl(5) for more information on Solaris ACL support.
```

Informationen zu Einschränkungen mit Zugriffskontrolllisten und Backup-Software finden Sie unter „Sichern von ZFS-Daten mit anderen Softwarepaketen zur Erstellung von Sicherungskopien“ auf Seite 229.

Syntaxbeschreibungen zum Setzen von Zugriffskontrolllisten

Es gibt zwei grundlegende Zugriffskontrolllistenformate:

- **Einfache Zugriffskontrollliste** – Enthält die herkömmlichen UNIX-Einträge `user`, `group` und `owner`.
- **Komplexe Zugriffskontrollliste** – Enthält mehr Einträge als nur "owner", "group" und "everyone" und enthält festgelegte Vererbungs-Flags oder Einträge, die nicht auf die herkömmliche Weise angeordnet sind.

Syntax zum Setzen gewöhnlicher Zugriffskontrolllisten

`chmod [Optionen] A[Index]{+|=}owner@ |group@ |everyone@:` *Zugriffsrechte*/...[:*Vererbungsflags*]: `deny | allow` *Datei*

`chmod [Optionen] A-owner@, group@, everyone@:` *Zugriffsrechte*/...[:*Vererbungsflags*]: `deny | allow` *Datei* ...

`chmod [Optionen] A[Index]-` *Datei*

Syntax zum Setzen komplexerer Zugriffskontrolllisten

`chmod [Optionen] A[Index]{+|=}user|group:name:`*Zugriffsrechte* /...[:*Vererbungsflags*]`:deny | allow` *Datei*

`chmod [Optionen] A-user|group:name:`*Zugriffsrechte* /...[:*Vererbungsflags*]`:deny | allow` *Datei* ...

`chmod [Optionen] A[Index]-` *Datei*

owner@, group@, everyone@
: Der *Zugriffskontrolllisten-Eintragstyp* für die gewöhnliche Zugriffskontrolllistensyntax. Eine Beschreibung der *Zugriffskontrolllisten-Eintragstypen* finden Sie in Tabelle 7–1.

user oder group:*Zugriffskontrolllisteneintrags-ID=Benutzername oder Gruppenname*
Der *Zugriffskontrolllisten-Eintragstyp* für die explizite Zugriffskontrolllistensyntax. Der *Zugriffskontrolllisten-Eintragstyp* "user" bzw. "group" muss auch die *Zugriffskontrolllisteneintrags-ID*, den *Benutzernamen* bzw. *Gruppennamen* enthalten. Eine Beschreibung der *Zugriffskontrolllisten-Eintragstypen* finden Sie in Tabelle 7–1.

Zugriffsrechte/.../
Die Zugriffsrechte, die gewährt bzw. verweigert werden. Eine Beschreibung von Zugriffsrechten für Zugriffskontrolllisten finden Sie in Tabelle 7–2.

Vererbungs-Flags
Eine optionale Liste von Vererbungsflags von Zugriffskontrolllisten. Eine Beschreibung der Vererbungsflags von Zugriffskontrolllisten finden Sie in Tabelle 7–4.

`deny | allow`
Legt fest, ob Zugriffsrechte gewährt oder verweigert werden.

Im folgenden Beispiel ist keine *Zugriffskontrolllisteneintrags-ID* für owner@, group@ oder everyone@ vorhanden.

```
group@:write_data/append_data/execute:deny
```

Das folgende Beispiel enthält eine *Zugriffssteuerungslisteneintrags-ID*, da ein spezifischer Benutzer (*Zugriffssteuerungslisten-Eintragstyp*) in der Zugriffssteuerungsliste enthalten ist.

```
0:user:gozer:list_directory/read_data/execute:allow
```

Ein Zugriffskontrolllisteneintrag sieht ungefähr wie folgt aus:

```
2:group@:write_data/append_data/execute:deny
```

Die **2** bzw. die *Index-ID* in diesem Beispiel identifiziert den Zugriffskontrolllisteneintrag in der größeren Zugriffskontrollliste, in der für Eigentümer, spezifische UIDs, Gruppen und allgemeine Zugriffsrechte mehrere Einträge vorhanden sein können. Sie können die *Index-ID* mit dem Befehl `chmod` angeben und somit festlegen, welchen Teil der Zugriffskontrollliste Sie ändern möchten. Sie können beispielsweise Index-ID 3 als `A3` im Befehl `chmod` angeben (siehe folgendes Beispiel):

```
chmod A3=user:venkman:read_acl:allow filename
```

Zugriffskontrolllisten-Eintragstypen, die Eigentümer, Gruppen und andere Parameter in Zugriffskontrolllisten darstellen, sind in der folgenden Tabelle beschrieben.

TABELLE 7–1 Zugriffskontrolllisten- Eintragstypen

Zugriffskontrolllisten-Eintragstyp	Beschreibung
`owner@`	Das Zugriffsrecht, das dem Eigentümer des Objekts gewährt wird.

TABELLE 7–1 Zugriffskontrolllisten- Eintragstypen *(Fortsetzung)*

Zugriffskontrolllisten-Eintragstyp	Beschreibung
`group@`	Das Zugriffsrecht, das der Eigentümergruppe des Objekts gewährt wird.
`everyone@`	Der Zugriff, der allen anderen Benutzern oder Gruppen gewährt wird, denen keine anderen Zugriffskontrolllisteneinträge entsprechen.
`user`	Das Zugriffsrecht, das einem zusätzlichen (und durch den Benutzernamen anzugebenden) Benutzer des Objekts gewährt wird. Muss die *Zugriffskontrolllisteneintrags-ID* enthalten, die wiederum den betreffenden *Benutzernamen* oder die *Benutzer-ID* enthält. Wenn es sich bei diesem Wert um keine gültige numerische Benutzer-ID oder keinen *Benutzername* handelt, ist der Zugriffskontrolllisten-Eintragstyp ungültig.
`group`	Das Zugriffsrecht, das einer zusätzlichen (und durch den Gruppennamen anzugebenden) Benutzergruppe des Objekts gewährt wird. Muss die *Zugriffskontrolllisteneintrags-ID* enthalten, die wiederum den betreffenden *Gruppennamen* oder die *Gruppen-ID* enthält. Wenn es sich bei diesem Wert um keine gültige numerische Gruppen-ID oder keinen *Gruppenname* handelt, ist der Zugriffskontrolllisten-Eintragstyp ungültig.

Zugriffssteuerungslisten-Zugriffsrechte sind in der folgenden Tabelle beschrieben.

TABELLE 7–2 Zugriffskontrolllisten-Zugriffsrechte

Zugriffsrecht	Kurzform	Beschreibung
`add_file`	`w`	Berechtigung zum Hinzufügen neuer Dateien zu einem Verzeichnis.
`add_subdirectory`	`p`	Berechtigung zum Erstellen von Unterverzeichnissen in einem Verzeichnis.
`append_data`	`p`	Gegenwärtig nicht implementiert.
`delete`	`d`	Berechtigung zum Löschen einer Datei. Weitere Informationen über das spezifische Verhalten des `delete`-Zugriffsrechts finden Sie in Tabelle 7–3.
`delete_child`	`D`	Berechtigung zum Löschen einer Datei bzw. eines Verzeichnisses in einem Verzeichnis. Weitere Informationen über das spezifische Verhalten des `delete_child`-Zugriffsrechts finden Sie in Tabelle 7–3.
`execute`	`x`	Berechtigung zum Ausführen einer Datei bzw. Durchsuchen eines Verzeichnisses.
`list_directory`	`r`	Berechtigung zum Auflisten des Inhalts eines Verzeichnisses.
`read_acl`	`c`	Berechtigung zum Lesen der Zugriffskontrollliste (`ls`).

TABELLE 7–2 Zugriffskontrolllisten-Zugriffsrechte *(Fortsetzung)*

Zugriffsrecht	Kurzform	Beschreibung
`read_attributes`	`a`	Berechtigung zum Lesen grundlegender (nicht zu Zugriffskontrolllisten gehörender) Attribute einer Datei. Grundlegende Attribute sind Attribute auf der stat-Ebene. Durch Setzen dieses Zugriffsmaskenbits kann ein Objekt `ls`(1) und `stat`(2) ausführen.
`read_data`	`r`	Berechtigung zum Lesen des Inhalts einer Datei.
`read_xattr`	`R`	Berechtigung zum Lesen der erweiterten Attribute einer Datei bzw. Durchsuchen des Verzeichnisses erweiterter Dateiattribute.
`synchronize`	`s`	Gegenwärtig nicht implementiert.
`write_xattr`	`W`	Berechtigung zum Erstellen erweiterter Attribute bzw. Schreiben in das Verzeichnis erweiterter Attribute. Durch Gewähren dieses Zugriffsrechtes kann ein Benutzer für eine Datei ein Verzeichnis erweiterter Attribute erstellen. Die Zugriffsrechte der Attributdatei legen das Zugriffsrecht des Benutzers auf das Attribut fest.
`write_data`	`w`	Berechtigung zum Ändern oder Ersetzens des Inhalts einer Datei.
`write_attributes`	`A`	Berechtigung zum Setzen der Zeitmarken einer Datei bzw. eines Verzeichnisses auf einen beliebigen Wert.
`write_acl`	`C`	Berechtigung zum Schreiben bzw. Ändern der Zugriffsteuerungliste mithilfe des Befehls `chmod`.
`write_owner`	`o`	Berechtigung zum Ändern des Eigentümers bzw. der Gruppe einer Datei bzw. Berechtigung zum Ausführen der Befehle `chown` oder `chgrp` für die betreffende Datei. Berechtigung zum Übernehmen der Eigentümerschaft einer Datei bzw. zum Ändern der ihrer Gruppe, zu der der betreffende Benutzer gehört. Wenn Sie die Benutzer- bzw. Gruppeneigentümerschaft auf einen beliebigen Benutzer bzw. eine beliebige Gruppe setzen wollen, ist das Zugriffsrecht `PRIV_FILE_CHOWN` erforderlich.

Die folgende Tabelle enthält zusätzliche Details zu dem Zugriffskontrolllistenverhalten `delete` und `delete_child`.

TABELLE 7–3 Verhalten der delete- und delete_child-Zugriffsrechte der Zugriffskontrollliste

Übergeordnete Verzeichniszugriffsrechte	Zielobjektzugriffsrechte		
	Zugriffskontrollliste lässt "delete" zu	Zugriffskontrollliste lässt "delete" nicht zu	Delete-Zugriffsrecht nicht angegeben
Zugriffskontrollliste lässt `delete_child` zu	Zulassen	Zulassen	Zulassen
Zugriffskontrollliste lässt `delete_child` nicht zu	Zulassen	Verweigern	Verweigern
Zugriffskontrollliste lässt nur `write` und `execute` zu	Zulassen	Zulassen	Zulassen
Zugriffskontrollliste lässt `write` und `execute` nicht zu	Zulassen	Verweigern	Verweigern

Vererbung von Zugriffskontrolllisten

Der Zweck der Vererbung von Zugriffssteuerungslisten besteht darin, dass neu erstellte Dateien bzw. Verzeichnisse die vorgesehenen Zugriffssteuerungslisten erben, ohne dass die vorhandenen Zugriffsrecht-Bits des übergeordneten Verzeichnisses ignoriert werden.

Standardmäßig werden Zugriffskontrolllisten nicht weitergegeben. Wenn Sie für ein Verzeichnis eine komplexe Zugriffskontrollliste setzen, wird diese nicht an untergeordnete Verzeichnisse vererbt. Sie müssen die Vererbung einer Zugriffskontrollliste für Dateien bzw. Verzeichnisse explizit angeben.

In der folgenden Tabelle sind die optionalen Vererbungsflags aufgeführt.

TABELLE 7–4 Vererbungs-Flags von Zugriffskontrolllisten

Vererbungs-Flag	Kurzform	Beschreibung
`file_inherit`	`f`	Die Zugriffskontrollliste wird nur vom übergeordneten Verzeichnis an die Dateien des betreffenden Verzeichnisses vererbt.
`dir_inherit`	`d`	Die Zugriffskontrollliste wird nur vom übergeordneten Verzeichnis an die untergeordneten Verzeichnisse des betreffenden Verzeichnisses vererbt.

TABELLE 7–4 Vererbungs-Flags von Zugriffskontrolllisten *(Fortsetzung)*

Vererbungs-Flag	Kurzform	Beschreibung
`inherit_only`	`i`	Die Zugriffskontrollliste wird vom übergeordneten Verzeichnis vererbt, gilt jedoch nur für neu erstellte Dateien bzw. Unterverzeichnisse, aber nicht für das betreffende Verzeichnis selbst. Für dieses Flag ist das Flag `file_inherit` bzw. `dir_inherit` (oder beide) erforderlich. Diese legen fest, was vererbt wird.
`no_propagate`	`n`	Die Zugriffskontrollliste wird vom übergeordneten Verzeichnis an die erste Hierarchieebene des betreffenden Verzeichnisses und nicht an untergeordnete Ebenen vererbt. Für dieses Flag ist das Flag `file_inherit` bzw. `dir_inherit` (oder beide) erforderlich. Diese legen fest, was vererbt wird.
-	entf.	Zugriff nicht gewährt.

Darüber hinaus können Sie mithilfe der Dateisystemeigenschaft `aclinherit` für das Dateisystem mehr oder weniger strenge Vererbungsrichtlinien für Zugriffskontrolllisten festlegen. Weitere Informationen finden Sie im folgenden Abschnitt.

Eigenschaften von Zugriffskontrolllisten

Das ZFS-Dateisystem umfasst die folgenden Eigenschaften von Zugriffskontrolllisten, um das spezifische Verhalten der Zugriffskontrolllistenvererbung und Zugriffskontrolllisteninteraktion mit `chmod`-Vorgängen zu bestimmen.

- `aclinherit` – Bestimmt das Verhalten der Zugriffskontrolllistenvererbung. Folgende Werte stehen zur Verfügung:
  - `discard` – Für neu erstellte Objekte werden beim Erstellen von Dateien und Verzeichnissen keine Zugriffskontrolllisteneinträge vererbt. Die Zugriffskontrollliste der Datei bzw. des Verzeichnisses entspricht dem Zugriffsrechtsmodus dieser Datei bzw. dieses Verzeichnisses.
  - `noallow` – Bei neuen Objekten werden nur vererbbare Zugriffssteuerungslisteneinträge mit Zugriffstyp `deny` vererbt.
  - `restricted` – Für neu erstellte Objekte werden beim Vererben von Zugriffskontrolllisteneinträgen die Zugriffsrechte `write_owner` und `write_acl` entfernt.
  - `passthrough` – Ist der Eigenschaftswert auf `passthrough` gesetzt, werden die Dateien mit einem durch die vererbbaren Zugriffskontrolleinträge festgelegten Modus erstellt. Wenn keine den Modus betreffenden vererbbaren Zugriffskontrolleinträge vorhanden sind, wird ein mit der Forderung der Anwendung vereinbarter Modus gesetzt.

- `passthrough-x` – Hat die gleiche Semantik wie `passthrough`, mit der Ausnahme, dass bei Aktivierung von `passthrough-x` Dateien mit der Ausführungsberechtigung (`x`) erstellt werden, jedoch nur, wenn die Ausführungsberechtigung im Dateierstellungsmodus und in eines vererbbaren Zugriffskontrolleintrags festgelegt wurde, die sich auf den Modus auswirkt.

Der Standardmodus für `aclinherit` ist `restricted`.

- `aclmode` – Ändert das Verhalten der Zugriffskontrollliste, wenn eine Datei das erste Mal erstellt wird, oder kontrolliert, wie eine Zugriffskontrollliste während eines `chmod`-Vorgangs geändert wird. Die Werte umfassen:
  - `discard` – Ein Dateisystem mit einer `aclmode`-Eigenschaft `discard` löscht alle Zugriffskontrolllisteneinträge, die den Modus der Datei nicht darstellen. Dies ist der Standardwert.
  - `mask` – Ein Dateisystem mit einer `aclmode` -Eigenschaft `mask` kürzt die Benutzer- oder Gruppenzugriffsrechte. Die Zugriffsrechte werden gekürzt, sodass sie nicht größer sind als die Gruppenzugriffsrechte-Bits, es sei denn, es handelt sich um einen Benutzereintrag, der dieselbe UID wie der Eigentümer der Datei oder des Verzeichnisses hat. In diesem Fall werden die Zugriffsrechte der Zugriffskontrollliste gekürzt, sodass sie nicht größer sind als die Eigentümerzugriffsrechte-Bits. Der Maskenwert behält die Zugriffskontrollliste auch über Modusänderungen hinweg bei, vorausgesetzt, es wurde kein expliziter Vorgang zur Festlegung der Zugriffskontrollliste ausgeführt.
  - `passthrough` – Ein Dateisystem mit einer `aclmode`-Eigenschaft `passthrough` gibt an, dass keine Änderungen an der Zugriffskontrollliste vorgenommen werden, außer der Generierung der erforderlichen Zugriffskontrolllisteneinträge zur Darstellung des neuen Modus der Datei oder des Verzeichnisses.

  Der Standardmodus für `aclmode` ist `discard`.

  Weitere Informationen zur Verwendung der Eigenschaft `aclmode` finden Sie in Beispiel 7–13.

Setzen von Zugriffskontrolllisten an ZFS-Dateien

In ZFS bestehen Zugriffskontrolllisten aus einer Folge von Zugriffskontrolllisteneinträgen. ZFS bietet ein *reines* Zugriffskontrolllistenmodell, in dem alle Dateien eine Zugriffskontrollliste besitzen. Normalerweise sind solche Zugriffssteuerungslisten *gewöhnlich*, da sie nur die herkömmlichen UNIX-Einträge `owner/group/other` repräsentieren.

ZFS-Dateien besitzen zwar trotzdem Zugriffsrecht-Bits und einen Zugriffsmodus files, diese stellen jedoch einen "Speicher" des Inhalts der entsprechenden Zugriffskontrollliste dar. Wenn Sie also die Zugriffsrechte einer Datei ändern, wird die entsprechende Zugriffskontrollliste automatisch aktualisiert. Darüber hinaus kann nach dem Entfernen einer komplexen

Zugriffskontrollliste, mit der ein Benutzer Zugriff auf eine Datei bzw. ein Verzeichnis hatte, wegen der Zugriffsrecht-Bits dieser Datei bzw. dieses Verzeichnisses, die einen Zugriff für alle Benutzer festlegten, dieser Benutzer unter Umständen noch immer Zugriff auf diese Datei bzw. dieses Verzeichnis haben. Alle Entscheidungen in Sachen Zugriffskontrolle werden von den Zugriffsrechten der Zugriffssteuerungsliste dieser Datei bzw. dieses Verzeichnisses festgelegt.

Die grundlegenden Regeln zum Zugriff auf ZFS-Dateien mithilfe von Zugriffskontrolllisten sind wie folgt:

- ZFS verarbeitet Zugriffskontrolllisteneinträge in der Reihenfolge, wie sie in der Zugriffskontrollliste aufgeführt sind (d. h. von oben nach unten).
- Es werden nur Zugriffskontrolllisteneinträge berücksichtigt, deren Benutzer-ID mit der ID des Zugriff anfordernden Benutzer übereinstimmt.
- Wenn ein Zugriffsrecht einmal gewährt wurde, kann es durch nachfolgende Zugriffskontrolllisteneinträge in der gleichen Zugriffskontrollliste nicht mehr rückgängig gemacht werden.
- Eigentümern von Dateien wird stets bedingungslose das Zugriffsrecht `write_acl` auch dann gewährt, wenn es in der Zugriffskontrollliste explizit verweigert wird. Nicht angegebene Zugriffsrechte werden verweigert.

  Falls Zugriffsrechte verweigert werden bzw. Angaben zum betreffenden Zugriffsrecht fehlen, legen die Zugriffsrechte des betreffenden untergeordneten Systems fest, welche Rechte dem Eigentümer einer Datei bzw. dem Superuser gewährt werden. Dadurch wird verhindert, dass der Zugriff für Dateieigentümernicht gesperrt wird und Superuser Dateien aus Gründen der Wiederherstellung ändern können.

Wenn Sie für ein Verzeichnis eine komplexe Zugriffskontrollliste setzen, wird diese nicht automatisch an untergeordnete Verzeichnisse dieses Verzeichnisses vererbt. Wenn Sie für ein Verzeichnis eine komplexe Zugriffskontrollliste setzen und diese an die untergeordneten Verzeichnisse dieses Verzeichnisses vererbt werden soll, müssen Sie die Zugriffskontrolllisten-Vererbungsflags verwenden. Weitere Informationen finden Sie in Tabelle 7–4 und unter „Festlegen der Vererbung von Zugriffskontrolllisten an ZFS-Dateien im Verbose-Format“ auf Seite 257.

Erstellen einer neuen Datei und (je nach dem Wert von `umask`) einer gewöhnlichen Standard-Zugriffssteuerungsliste ähnlich wie im folgenden Beispiel:

```
$ ls -v file.1
-rw-r--r--   1 root     root      206663 Jun 23 15:06 file.1
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

In diesem Beispiel besitzt jede Benutzerkategorie (`owner@`, `group@`, `everyone@`) einen Zugriffskontrolllisteneintrag.

Im Folgenden finden Sie eine Beschreibung dieser Datei-Zugriffssteuerungsliste:

`0:owner@` Der Eigentümer kann den Dateiinhalt lesen und ändern (`read_data/write_data/append_data/read_xattr`). Der Eigentümer kann darüber hinaus auch Dateiattribute wie z. B. Zeitstempel, erweiterte Attribute und Zugriffskontrolllisten ändern (`write_xattr/read_attributes/write_attributes/ read_acl/write_acl`). Zusätzlich dazu kann der Eigentümer die Datei-Eigentümerschaft ändern (`write_owner:allow`).

Das Zugriffsrecht `synchronize` ist zurzeit nicht implementiert.

`1:group@` Der Gruppe wird die Berechtigung zum Lesen der Datei bzw. der Dateiattribute gewährt (`read_data/read_xattr/read_attributes/read_acl:allow`).

`2:everyone@` Allen übrigen Benutzern und Gruppen wird die Berechtigung zum Lesen der Datei bzw. der Dateiattribute gewährt (`read_data/read_xattr/read_attributes/read_acl/ synchronize:allow`). Das Zugriffsrecht `synchronize` ist zurzeit nicht implementiert.

Erstellen eines neuen Verzeichnisses und (je nach dem Wert von `umask`) einer standardmäßigen Verzeichnis-Zugriffskontrollliste ähnlich wie im folgenden Beispiel:

```
$ ls -dv dir.1
drwxr-xr-x   2 root     root           2 Jul 20 13:44 dir.1
     0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Im Folgenden finden Sie eine Beschreibung dieser Verzeichnis-Zugriffskontrollliste:

`0:owner@` Der Eigentümer kann den Inhalt des Verzeichnisses lesen und ändern (`list_directory/read_data/add_file/write_data/add_subdirectory /append_data`) und die Dateiattribute wie z. B. Zeitstempel, erweiterte Attribute und Zugriffskontrolllisten lesen und ändern (`/read_xattr/write_xattr/read_attributes/write_attributes/read_acl/ write_acl`). Außerdem kann der Eigentümer den Inhalt durchsuchen (`execute`), eine Datei oder ein Verzeichnis löschen (`delete_child`) und die Zugehörigkeit des Verzeichnisses (`write_owner:allow`) ändern.

Das Zugriffsrecht `synchronize` ist zurzeit nicht implementiert.

`1:group@` Die Gruppe kann den Inhalt und die Attribute des Verzeichnisses auflisten und lesen. Außerdem kann die Gruppe den Verzeichnisinhalt durchsuchen (`list_directory/read_data/read_xattr/execute/read_attributes /read_acl/synchronize:allow`).

`2:everyone@` Allen übrigen Benutzern und Gruppen wird die Berechtigung zum Lesen und Durchsuchen des Verzeichnisinhalts und der Verzeichnisattribute gewährt (`list_directory/read_data/read_xattr/execute/read_ attributes/read_acl/synchronize:allow`). Das Zugriffsrecht `synchronize` ist zurzeit nicht implementiert.

Setzen und Anzeigen von Zugriffskontrolllisten an ZFS-Dateien im Verbose-Format

Mit dem Befehl `chmod` können Sie Zugriffskontrolllisten von ZFS-Dateien ändern. Die folgende `chmod`-Syntax zum Ändern von Zugriffskontrolllisten nutzt zur Erkennung des Zugriffskontrolllistenformats die *Zugriffskontrolllisten-Spezifikation.* Eine Beschreibung der *Zugriffskontrolllisten-Spezifikation* finden Sie unter „Syntaxbeschreibungen zum Setzen von Zugriffskontrolllisten“ auf Seite 243.

- Hinzufügen von Zugriffskontrolllisten-Eintragstypen
  - Hinzufügen eines Zugriffskontrolllisteneintrags für einen Benutzer

    ```
    % chmod A+acl-specification filename
    ```
  - Hinzufügen eines Zugriffskontrolllisteneintrags nach *Index-ID*

    ```
    % chmod Aindex-ID+acl-specification filename
    ```

    Mit dieser Syntax wird der neue Zugriffskontrolllisteneintrag an der entsprechenden *Index-ID* eingefügt.
- Ersetzen eines Zugriffskontrolllisteneintrags

  ```
  % chmod A=acl-specification filename
  ```

  ```
  % chmod Aindex-ID=acl-specification filename
  ```
- Entfernen von Zugriffskontrolllisteneinträgen
  - Entfernen eines Zugriffskontrolllisteneintrags nach *Index-ID*

    ```
    % chmod Aindex-ID- filename
    ```
  - Entfernen eines Zugriffskontrolllisteneintrags nach Benutzername

    ```
    % chmod A-acl-specification filename
    ```
  - Entfernen aller komplexen Zugriffskontrolleinträge aus einer Datei

    ```
    % chmod A- filename
    ```

Ausführliche Informationen zu Zugriffskontrolllisten werden mithilfe des Befehls `ls -v` angezeigt. Beispiel:

```
# ls -v file.1
-rw-r--r--   1 root     root      206695 Jul 20 13:43 file.1
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Informationen zur Verwendung des Zugriffskontrolllisten-Kompaktformats finden Sie unter „Setzen und Anzeigen von Zugriffskontrolllisten an ZFS-Dateien im Kompaktformat“ auf Seite 263.

BEISPIEL 7–1 Ändern gewöhnlicher Zugriffskontrolllisten an ZFS-Dateien

Dieser Abschnitt enthält Beispiele für die Festlegung und Anzeige von einfachen Zugriffskontrolllisten, d.h. nur die herkömmlichen UNIX-Einträge "user", "group" und "other" sind in der Zugriffskontrollliste enthalten.

Im folgenden Beispiel besitzt `Datei.1` eine gewöhnliche Zugriffskontrollliste:

```
# ls -v file.1
-rw-r--r--   1 root     root      206695 Jul 20 13:43 file.1
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Im folgenden Beispiel wird die Berechtigung `write_data` für `group@` gewährt.

```
# chmod A1=group@:read_data/write_data:allow file.1
# ls -v file.1
-rw-rw-r--   1 root     root      206695 Jul 20 13:43 file.1
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/write_data:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Im folgenden Beispiel werden die Berechtigungen an `Datei.1` auf 644 zurückgesetzt.

```
# chmod 644 file.1
# ls -v file.1
-rw-r--r--   1 root     root      206695 Jul 20 13:43 file.1
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
```

BEISPIEL 7–1 Ändern gewöhnlicher Zugriffskontrolllisten an ZFS-Dateien *(Fortsetzung)*

```
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

BEISPIEL 7–2 Setzen komplexer Zugriffskontrolllisten an ZFS-Dateien

Dieser Abschnitt enthält Beispiele zum Setzen und Anzeigen komplexer Zugriffskontrolllisten.

Im folgenden Beispiel werden die Berechtigungen `read_data/execute` für den Benutzer `gozer` und das•Verzeichnis `Test.Verz` gesetzt.

```
# chmod A+user:gozer:read_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:23 test.dir
     0:user:gozer:list_directory/read_data/execute:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Im folgenden Beispiel werden die Berechtigungen `read_data/execute` für den Benutzer `gozer` entfernt.

```
# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x   2 root     root           2 Jul 20 14:23 test.dir
     0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

BEISPIEL 7–3 Interaktion von Zugriffskontrolllisten mit Berechtigungen an ZFS-Dateien

Die folgenden Beispiele demonstrieren die Interaktion zwischen dem Setzen von Zugriffskontrolllisten und dem anschließenden Ändern der Berechtigungs-Bits einer Datei bzw. eines Verzeichnisses.

Im folgenden Beispiel besitzt `Datei.2` eine gewöhnliche Zugriffskontrollliste:

```
# ls -v file.2
-rw-r--r--   1 root     root        2693 Jul 20 14:26 file.2
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
```

BEISPIEL 7–3 Interaktion von Zugriffskontrolllisten mit Berechtigungen an ZFS-Dateien *(Fortsetzung)*

```
        /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
        :allow
```

Im folgenden Beispiel werden gewährte (`allow`) Zugriffskontrolllistenberechtigungen vom Benutzer `everyone@` entfernt.

```
# chmod A2- file.2
# ls -v file.2
-rw-r-----   1 root     root        2693 Jul 20 14:26 file.2
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
        /read_attributes/write_attributes/read_acl/write_acl/write_owner
        /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
```

In dieser Ausgabe werden die Dateiberechtigungs-Bits von 644 auf 640 zurückgesetzt. Leseberechtigungen für `everyone@` werden faktisch von den Dateiberechtigungs-Bits entfernt, wenn die gewährten Zugriffssteuerungslisten-Zugriffsrechte für `everyone@` entfernt werden.

Im folgenden Beispiel wird die vorhandene Zugriffskontrollliste mit den Zugriffsrechten `read_data/write_data` für `everyone@` ersetzt.

```
# chmod A=everyone@:read_data/write_data:allow file.3
# ls -v file.3
-rw-rw-rw-   1 root     root        2440 Jul 20 14:28 file.3
     0:everyone@:read_data/write_data:allow
```

In dieser Ausgabe ersetzt die `chmod`-Syntax faktisch die Zugriffskontrollliste mit den Berechtigungen `read_data/write_data:allow` durch Lese- und Schreibberechtigungen für Eigentümer, Gruppen und `everyone@`. In diesem Modell gibt `everyone@` den Zugriff für beliebige Benutzer bzw. Gruppen an. Da keine Zugriffskontrolllisteneinträge für `owner@` bzw. `group@` existieren, die die Zugriffsrechte für Eigentümer und Gruppen überschreiben könnten, werden die Berechtigungs-Bits auf 666 gesetzt.

Im folgenden Beispiel wird die vorhandene Zugriffskontrollliste mit Leseberechtigungen für den Benutzer `gozer` ersetzt.

```
# chmod A=user:gozer:read_data:allow file.3
# ls -v file.3
----------+  1 root     root        2440 Jul 20 14:28 file.3
     0:user:gozer:read_data:allow
```

In diesem Beispiel werden die Dateizugriffsrechte mit 000 berechnet, da für `owner@`, `group@` bzw. `everyone@` keine Zugriffskontrolllisteneinträge vorhanden sind, die die herkömmlichen Berechtigungskomponenten einer Datei repräsentieren. Der Dateieigentümer kann dieses Problem durch Zurücksetzen der Zugriffsrechte (und der Zugriffskontrollliste) wie folgt beheben:

BEISPIEL 7–3 Interaktion von Zugriffskontrolllisten mit Berechtigungen an ZFS-Dateien *(Fortsetzung)*

```
# chmod 655 file.3
# ls -v file.3
-rw-r-xr-x   1 root     root        2440 Jul 20 14:28 file.3
     0:owner@:execute:deny
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/execute/read_attributes/read_acl
         /synchronize:allow
     3:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
         /synchronize:allow
```

BEISPIEL 7–4 Wiederherstellen gewöhnlicher Zugriffskontrolllisten an ZFS-Dateien

Mit dem Befehl `chmod` können Sie alle komplexen Zugriffskontrolllisten einer Datei bzw. eines Verzeichnisses entfernen.

Im folgenden Beispiel besitzt `test5.dir` zwei komplexe Zugriffskontrolleinträge.

```
# ls -dv test5.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:32 test5.dir
     0:user:lp:read_data:file_inherit:deny
     1:user:gozer:read_data:file_inherit:deny
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     3:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     4:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Im folgenden Beispiel werden die komplexen Zugriffsteuerungslisten für die Benutzer `gozer` und `lp` entfernt. Die verbleibende Zugriffskontrollliste enthält die Standardwerte `owner@`, `group@` und `everyone@`.

```
# chmod A- test5.dir
# ls -dv test5.dir
drwxr-xr-x   2 root     root           2 Jul 20 14:32 test5.dir
     0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Festlegen der Vererbung von Zugriffskontrolllisten an ZFS-Dateien im Verbose-Format

Sie können festlegen, ob Zugriffsteuerungslisten von Dateien und Verzeichnissen vererbt werden sollen oder nicht. Standardmäßig werden Zugriffskontrolllisten nicht weitergegeben. Wenn Sie für ein Verzeichnis eine komplexe Zugriffskontrollliste setzen, wird diese nicht an untergeordnete Verzeichnisse vererbt. Sie müssen die Vererbung einer Zugriffskontrollliste für Dateien bzw. Verzeichnisse explizit angeben.

Die Eigenschaft `aclinherit` kann in einem Datensystem global gesetzt werden. Standardmäßig ist `aclinherit` auf `restricted` gesetzt.

Weitere Informationen dazu finden Sie in „Vererbung von Zugriffskontrolllisten“ auf Seite 247.

BEISPIEL 7–5 Gewähren der Standardvererbung von Zugriffskontrolllisten

Standardmäßig werden Zugriffskontrolllisten nicht durch eine Verzeichnisstruktur weitergegeben.

Im folgenden Beispiel wird ein komplexer Zugriffskontrolleintrag mit den Zugriffsrechten `read_data/write_data/execute` für den Benutzer `gozer` und das•Verzeichnis `test.dir` gesetzt.

```
# chmod A+user:gozer:read_data/write_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:53 test.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Beim Erstellen eines Unterverzeichnisses in `test.dir` wird der Zugriffskontrolleintrag für den Benutzer `gozer` nicht weitergegeben. Der Benutzer `gozer` hätte nur dann Zugriff auf das Unterverzeichnis `sub.dir`, wenn ihm die Berechtigungen für `sub.dir` Zugriff als Dateieigentümer, Gruppenmitglied oder `everyone@` gewähren würden.

```
# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x   2 root     root           2 Jul 20 14:54 test.dir/sub.dir
     0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

BEISPIEL 7–5 Gewähren der Standardvererbung von Zugriffskontrolllisten *(Fortsetzung)*

```
     2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

BEISPIEL 7–6 Gewähren der Vererbung von Zugriffskontrolllisten an Dateien und Verzeichnissen

In dieser Beispielfolge sind die Zugriffssteuerungslisten für Dateien und Verzeichnisse aufgeführt, die beim Setzen des Flags `file_inherit` angewendet werden.

Im folgenden Beispiel werden die Zugriffsrechte `read_data/write_data` für den Benutzer `gozer` des Verzeichnisses `test2.dir` hinzugefügt, sodass dieser Benutzer Leseberechtigung für neu erstellte Dateien besitzt.

```
# chmod A+user:gozer:read_data/write_data:file_inherit:allow test2.dir
# ls -dv test2.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:55 test2.dir
     0:user:gozer:read_data/write_data:file_inherit:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Im folgenden Beispiel werden die Zugriffsrechte des Benutzers `gozer` auf die neu erstellte Datei `test2.dir/file.2` angewendet. Durch die gewährte Zugriffskontrolllisten-Vererbung `read_data:file_inherit:allow` hat der Benutzer `gozer` Leseberechtigung für den Inhalt neu erstellter Dateien.

```
# touch test2.dir/file.2
# ls -v test2.dir/file.2
-rw-r--r--+  1 root     root           0 Jul 20 14:56 test2.dir/file.2
     0:user:gozer:read_data:inherited:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Da die Eigenschaft `aclinherit` für dieses Dateisystem auf den Standardwert `restricted` gesetzt ist, hat der Benutzer `gozer` für die Datei `file.2` nicht das Zugriffsrecht `write_data`, da es die Gruppenberechtigung der Datei nicht zulässt.

Bitte beachten Sie, dass das Zugriffsrecht `inherit_only`, das beim Setzen der Flags `file_inherit` bzw. `dir_inherit` angewendet wird, zum Weitergeben der Zugriffskontrollliste

BEISPIEL 7–6 Gewähren der Vererbung von Zugriffskontrolllisten an Dateien und Verzeichnissen *(Fortsetzung)*

durch die Verzeichnisstruktur dient. Somit werden dem Benutzer `gozer` nur Zugriffsrechte für `everyone@` gewährt bzw. verweigert, wenn er nicht Eigentümer der betreffenden Datei ist bzw. nicht zur Eigentümergruppe gehört. Beispiel:

```
# mkdir test2.dir/subdir.2
# ls -dv test2.dir/subdir.2
drwxr-xr-x+  2 root     root           2 Jun 23 15:21 test2.dir/subdir.2
     0:user:gozer:list_directory/read_data/add_file/write_data:file_inherit
         /inherit_only:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/read_attributes
         /write_attributes/read_acl/write_acl/write_owner/synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

In der folgenden Beispielfolge sind die Zugriffskontrolllisten für Dateien und Verzeichnisse aufgeführt, die beim Setzen des Flags `file_inherit` bzw. `dir_inherit` angewendet werden.

Im folgenden Beispiel werden dem Benutzer `gozer` Lese-, Schreib- und Ausführungsberechtigungen gewährt, die für neu erstellte Dateien und Verzeichnisse vererbt wurden.

```
# 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 Jul 20 15:00 test3.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
         :file_inherit/dir_inherit:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3: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 Jun 23 15:25 test3.dir/file.3
     0:user:gozer:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

BEISPIEL 7–6 Gewähren der Vererbung von Zugriffskontrolllisten an Dateien und Verzeichnissen *(Fortsetzung)*

```
# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+  2 root     root           2 Jun 23 15:26 test3.dir/subdir.1
     0:user:gozer:list_directory/read_data/execute:file_inherit/dir_inherit
         :allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/read_attributes
         /write_attributes/read_acl/write_acl/write_owner/synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Da die Zugriffsrecht-Bits des übergeordneten Verzeichnisses für `group@` und `everyone@` Schreib- und Ausführungsberechtigungen verweigern, wird in den obigen Beispielen dem Benutzer `gozer` ebenfalls die Schreib- und Ausführungsberechtigung verweigert. Der Standardwert der Eigenschaft `aclinherit` ist `restricted`, was bedeutet, dass die Zugriffsrechte `write_data` und `execute` nicht vererbt werden.

Im folgenden Beispiel werden dem Benutzer `gozer` Lese-, Schreib- und Ausführungsberechtigungen gewährt, die für neu erstellte Dateien und Verzeichnisse vererbt wurden, aber nicht an die untergeordnete Dateien des Verzeichnisses weitergegeben werden.

```
# chmod A+user:gozer:read_data/write_data/execute:file_inherit/no_propagate:allow
test4.dir
# ls -dv test4.dir
drwxr--r--+  2 root     root           2 Mar  1 12:11 test4.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
         :file_inherit/no_propagate:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/read_attributes/read_acl
         /synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
         /synchronize:allow
```

Wie das folgende Beispiel zeigt, werden die Zugriffsrechte `read_data/write_data/execute` des Benutzers `gozer` basierend auf den Rechten der Eigentümergruppe reduziert.

```
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+  1 root     root           0 Jun 23 15:28 test4.dir/file.4
     0:user:gozer:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
```

BEISPIEL 7–6 Gewähren der Vererbung von Zugriffskontrolllisten an Dateien und Verzeichnissen *(Fortsetzung)*

```
         :allow
```

BEISPIEL 7–7 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllisten-Vererbungsmodus "Pass Through"

Wenn die Eigenschaft `aclinherit` des Dateisystems `tank/cindy` auf `passthrough` gesetzt ist, erbt der Benutzer `gozer` die auf `test4.dir` angewendete Zugriffskontrollliste für die neu erstellte Datei `file.5` wie folgt:

```
# zfs set aclinherit=passthrough tank/cindy
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+  1 root     root           0 Jun 23 15:35 test4.dir/file.4
     0:user:gozer:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

BEISPIEL 7–8 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllistenvererbungsmodus "Discard"

Wenn die Eigenschaft `aclinherit` eines Dateisystems auf `discard` gesetzt ist, können Zugriffskontrolllisten potenziell ignoriert werden, wenn sich die Zugriffsrecht-Bits eines Verzeichnisses ändern. Beispiel:

```
# zfs set aclinherit=discard tank/cindy
# 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 Jul 20 14:18 test5.dir
     0:user:gozer:list_directory/read_data/add_file/write_data/execute
         :dir_inherit:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Wenn Sie später die Zugriffsrechte an einem Verzeichnis einschränken, gelten die komplexen Zugriffskontrolllisten nicht mehr. Beispiel:

```
# chmod 744 test5.dir
# ls -dv test5.dir
drwxr--r--   2 root     root           2 Jul 20 14:18 test5.dir
     0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
```

BEISPIEL 7–8 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllistenvererbungsmodus "Discard" *(Fortsetzung)*

```
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:list_directory/read_data/read_xattr/read_attributes/read_acl
         /synchronize:allow
     2:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
         /synchronize:allow
```

BEISPIEL 7–9 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllistenmodus "Noallow"

Im folgenden Beispiel sind zwei komplexe Zugriffsteuerungslisten mit Dateivererbung gesetzt. Eine Zugriffskontrollliste gewährt die Berechtigung `read_data`, und die andere Zugriffskontrollliste verweigert die Berechtigung `read_data`. Dieses Beispiel zeigt auch, wie Sie mit dem gleichen Aufruf des Befehls `chmod` zwei Zugriffskontrolleinträge angeben können.

```
# zfs set aclinherit=noallow tank/cindy
# 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 Jul 20 14:22 test6.dir
     0:user:gozer:read_data:file_inherit:deny
     1:user:lp:read_data:file_inherit:allow
     2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     3:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     4:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

Wie das folgende Beispiel zeigt, gilt beim Erstellen einer neuen Datei die Zugriffskontrollliste, die das Zugriffsrecht `read_data` gewährt, nicht mehr.

```
# touch test6.dir/file.6
# ls -v test6.dir/file.6
-rw-r--r--+  1 root     root           0 Jun 15 12:19 test6.dir/file.6
     0:user:gozer:read_data:inherited:deny
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Setzen und Anzeigen von Zugriffskontrolllisten an ZFS-Dateien im Kompaktformat

Sie können Zugriffsrechte an ZFS-Dateien in einem Kompaktformat setzen und anzeigen, das für die einzelnen Berechtigungen 14 eindeutige Buchstaben verwendet. Die Buchstaben, die die Zugriffsrechte in Kompaktform darstellen, werden in Tabelle 7–2 und Tabelle 7–4 aufgeführt.

Kompaktausgaben von Zugriffskontrolllisten für Dateien und Verzeichnisse werden mithilfe des Befehls `ls -V` angezeigt. Beispiel:

```
# ls -V file.1
-rw-r--r--   1 root     root      206663 Jun 23 15:06 file.1
            owner@:rw-p--aARWcCos:------:allow
            group@:r-----a-R-c--s:------:allow
         everyone@:r-----a-R-c--s:------:allow
```

Die Bedeutung dieser Kompaktausgabe ist wie folgt:

`owner@` Der Eigentümer kann den Dateiinhalt lesen und ändern (`rw=read_data/write_data`), (`p= append_data`). Der Eigentümer kann darüber hinaus auch Dateiattribute wie z. B. Zeitstempel, erweiterte Attribute und Zugriffskontrolllisten ändern (`a=read_attributes` , `W=write_xattr`, `R=read_xattr`, `A=write_attributes`, `c=read_acl`, `C=write_acl`). Zusätzlich dazu kann der Eigentümer die Datei-Eigentümerschaft ändern (`o=write_owner`).

Das Zugriffsrecht `synchronize` (`s`) ist aktuell nicht implementiert.

`group@` Der Gruppe wird die Berechtigung zum Lesen der Datei (`r= read_data`) bzw. der Dateiattribute (`a=read_attributes` , `R=read_xattr`, `c= read_acl`) gewährt.

Das Zugriffsrecht `synchronize` (`s`) ist aktuell nicht implementiert.

`everyone@` Allen übrigen Benutzern und Gruppen wird die Berechtigung zum Lesen der Datei bzw. der Dateiattribute gewährt (`r=read_data`, `a=append_data`, `R=read_xattr` , `c=read_acl` und `s= synchronize`).

Das Zugriffsrecht `synchronize` (`s`) ist aktuell nicht implementiert.

Das Kompaktformat von Zugriffskontrolllisten hat gegenüber dem Verbose-Format folgende Vorteile:

- Zugriffsrechte können für den Befehl `chmod` als positionale Argumente angegeben werden.
- Der Bindestrich (-), der für die Verweigerung von Zugriffsrechten steht, kann weggelassen werden, und es müssen nur die erforderlichen Buchstaben angegeben werden.
- Zugriffsrechte und Vererbungsflags werden in der gleichen Weise gesetzt.

Informationen zur Verwendung des Verbose-Formats von Zugriffskontrolllisten finden Sie unter „Setzen und Anzeigen von Zugriffskontrolllisten an ZFS-Dateien im Verbose-Format“ auf Seite 252.

BEISPIEL 7–10 Setzen und Anzeigen von Zugriffskontrolllisten im Kompaktformat

Im folgenden Beispiel besitzt `file.1` eine gewöhnliche Zugriffssteuerungsliste:

```
# ls -V file.1
-rw-r--r--   1 root     root      206663 Jun 23 15:06 file.1
           owner@:rw-p--aARWcCos:------:allow
           group@:r-----a-R-c--s:------:allow
        everyone@:r-----a-R-c--s:------:allow
```

In diesem Beispiel werden die Berechtigungen `read_data/execute` für den Benutzer `gozer` an der Datei `file.1` hinzugefügt.

```
# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+  1 root     root      206663 Jun 23 15:06 file.1
       user:gozer:r-x-----------:------:allow
           owner@:rw-p--aARWcCos:------:allow
           group@:r-----a-R-c--s:------:allow
        everyone@:r-----a-R-c--s:------:allow
```

Im folgenden Beispiel werden dem Benutzer `gozer` unter Verwendung des Zugriffskontrolllisten-Kompaktformats Lese-, Schreib- und Ausführungsberechtigungen gewährt, die für neu erstellte Dateien und Verzeichnisse vererbt wurden.

```
# chmod A+user:gozer:rwx:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+  2 root     root           2 Jun 23 16:04 dir.2
       user:gozer:rwx-----------:fd----:allow
           owner@:rwxp--aARWcCos:------:allow
           group@:r-x---a-R-c--s:------:allow
        everyone@:r-x---a-R-c--s:------:allow
```

Sie können Zugriffsrechte und Vererbungsflags auch aus der Ausgabe des Befehls `ls -V` in das Kompaktformat des Befehls `chmod` kopieren. Beispiel: Wenn Sie die Zugriffsrechte und Vererbungs-Flags von `dir.2` für den Benutzer `gozer` auf den Benutzer `cindy` an `dir.2` übertragen möchten, müssen Sie die entsprechenden Zugriffsrechte und Vererbungs-Flags (`rwx-----------:fd----:allow`) nur in die Befehlszeile des Befehls `chmod` kopieren. Beispiel:

```
# chmod A+user:cindy:rwx-----------:fd----:allow dir.2
# ls -dV dir.2
drwxr-xr-x+  2 root     root           2 Jun 23 16:04 dir.2
      user:cindy:rwx-----------:fd----:allow
       user:gozer:rwx-----------:fd----:allow
           owner@:rwxp--aARWcCos:------:allow
           group@:r-x---a-R-c--s:------:allow
        everyone@:r-x---a-R-c--s:------:allow
```

BEISPIEL 7–11 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllisten-Vererbungsmodus "Pass Through"

Wenn die Eigenschaft `aclinherit` des Dateisystems auf `passthrough` gesetzt ist, werden alle vererbbaren Zugriffskontrolllisten ohne jegliche Änderung der Einträge bei der Vererbung weitergegeben. Ist diese Eigenschaft auf `passthrough` gesetzt, werden Dateien mit einem Berechtigungsmodus erstellt, der vom vererbbaren Zugriffskontrolleintrag bestimmt wird. Wenn keine den Berechtigungsmodus betreffenden vererbbaren Zugriffskontrolleinträge vorhanden sind, wird ein mit der Forderung der Anwendung vereinbarter Berechtigungsmodus gesetzt.

In den folgenden Beispielen wird die Vererbung von Berechtigungs-Bits durch das Setzen des Modus `aclinherit` auf `passthrough` in kompakter Zugriffskontrolllistensyntax veranschaulicht.

In diesem Beispiel wird eine Zugriffskontrollliste für `test1.dir` gesetzt, um die die Vererbung zu erzwingen. Durch die Syntax wird für neu erstellte Dateien ein Zugriffskontrolleintrag `owner@`, `group@` und `everyone@` erstellt. Neu erstellte Verzeichnisse erben die Zugriffskontrolleinträge `@owner`, `group@` und `everyone@`.

```
# zfs set aclinherit=passthrough tank/cindy
# pwd
/tank/cindy
# 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 23 16:10 test1.dir
            owner@:rwxpdDaARWcCos:fd----:allow
            group@:rwxp----------:fd----:allow
         everyone@:--------------:fd----:allow
```

In diesem Beispiel erbt eine neu erstellte Datei die Zugriffssteuerungsliste, die für die Weitergabe an neu erstellte Dateien angegeben wurde.

```
# cd test1.dir
# touch file.1
# ls -V file.1
-rwxrwx---+  1 root     root           0 Jun 23 16:11 file.1
            owner@:rwxpdDaARWcCos:------:allow
            group@:rwxp----------:------:allow
         everyone@:--------------:------:allow
```

In diesem Beispiel erbt ein neu erstelltes Verzeichnis sowohl die Zugriffssteuerungseinträge, die den Zugriff auf dieses Verzeichnis regeln, als auch diejenigen für die künftige Weitergabe an untergeordnete Objekte des neu erstellten Verzeichnisses.

```
# mkdir subdir.1
# ls -dV subdir.1
```

BEISPIEL 7–11 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllisten-Vererbungsmodus "Pass Through" *(Fortsetzung)*

```
drwxrwx---+  2 root     root           2 Jun 23 16:13 subdir.1
            owner@:rwxpdDaARWcCos:fd----:allow
            group@:rwxp----------:fd----:allow
         everyone@:--------------:fd----:allow
```

Die Einträge `fd----` gelten für die Weitergabe der Vererbung und werden bei der Zugriffskontrolle nicht berücksichtigt. In diesem Beispiel wird eine Datei mit einer gewöhnlichen Zugriffssteuerungsliste in einem anderen Verzeichnis erstellt, wo keine vererbten Zugriffssteuerungseinträge vorhanden sind.

```
# cd /tank/cindy
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r--   1 root     root           0 Jun 23 16:15 file.2
            owner@:rw-p--aARWcCos:------:allow
            group@:r-----a-R-c--s:------:allow
         everyone@:r-----a-R-c--s:------:allow
```

BEISPIEL 7–12 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllisten-Vererbungsmodus "Pass Through-X"

Ist `aclinherit=passthrough-x` aktiviert, so werden Dateien mit der Ausführungsberechtigung (`x`) für `owner@`, `group@` oder `everyone@` erstellt, allerdings nur, wenn die Ausführungsberechtigung im Dateierstellungsmodus und in einem vererbbaren Zugriffskontrolleintrag, der den Modus betrifft, eingestellt ist.

Das folgende Beispiel zeigt, wie die Ausführungsberechtigung durch Einstellen des Modus `aclinherit` auf `passthrough-x` vererbt wird.

```
# zfs set aclinherit=passthrough-x tank/cindy
```

Die folgende Zugriffskontrollliste ist auf `/tank/cindy/test1.dir` gesetzt, um ausführbare Zugriffskontrolllisten-Vererbung für Dateien für `owner@` bereitzustellen.

```
# 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 23 16:17 test1.dir
            owner@:rwxpdDaARWcCos:fd----:allow
            group@:rwxp----------:fd----:allow
         everyone@:--------------:fd----:allow
```

Es wird eine Datei (`file1`) mit den angeforderten Zugriffsrechten `0666` erstellt. Die resultierenden Zugriffsrechte sind `0660`. Die Ausführungsberechtigung wurde nicht vererbt, weil der Erstellungsmodus dies nicht fordert.

BEISPIEL 7–12 Vererbung von Zugriffskontrolllisten mit dem Zugriffskontrolllisten-Vererbungsmodus "Pass Through-X" *(Fortsetzung)*

```
# touch test1.dir/file1
# ls -V test1.dir/file1
-rw-rw----+  1 root     root           0 Jun 23 16:18 test1.dir/file1
            owner@:rw-pdDaARWcCos:------:allow
            group@:rw-p----------:------:allow
         everyone@:--------------:------:allow
```

Als Nächstes wird das Executable `t` unter Verwendung des Compilers `cc` im Verzeichnis `testdir` erstellt.

```
# cc -o t t.c
# ls -V t
-rwxrwx---+  1 root     root        7396 Dec  3 15:19 t
                owner@:rwxpdDaARWcCos:------:allow
                group@:rwxp----------:------:allow
              everyone@:--------------:------:allow
```

Die resultierenden Zugriffsrechte sind `0770`, weil `cc` die Zugriffsrechte `0777` gefordert hat, weswegen die Ausführungsberechtigung von den Einträgen `owner@`, `group@` und `everyone@` geerbt wurde.

BEISPIEL 7–13 Interaktion der Zugriffskontrollliste mit `chmod`-Vorgängen in ZFS-Dateien

In den folgenden Beispielen wird dargestellt, wie sich bestimmte `aclmode`- und `aclinherit`-Eigenschaftswerte auf die Interaktion der vorhandenen Zugriffskontrolllisten mit einem `chmod`-Vorgang auswirken, der Datei- oder Verzeichniszugriffsrechte ändert, um vorhandene Zugriffsrechte der Zugriffskontrolllisten zu verringern oder zu erweitern, damit sie konsistent mit der Eigentümergruppe sind.

In diesem Beispiel ist die `aclmode`-Eigenschaft auf `mask` und die `aclinherit`-Eigenschaft auf `restricted` gesetzt. Die Zugriffsrechte der Zugriffskontrollliste in diesem Beispiel werden in Kompaktform angezeigt, in der die Änderung der Zugriffsrechte einfacher dargestellt werden kann.

Die ursprüngliche Datei- und Gruppeneigentümerschaft und die Zugriffsrechte der Zugriffskontrollliste sind:

```
# zfs set aclmode=mask pond/whoville
# zfs set aclinherit=restricted pond/whoville

# ls -lV file.1
-rwxrwx---+  1 root     root      206695 Aug 30 16:03 file.1
               user:amy:r-----a-R-c---:-------:allow
              user:rory:r-----a-R-c---:-------:allow
         group:sysadmin:rw-p--aARWc---:-------:allow
            group:staff:rw-p--aARWc---:-------:allow
                 owner@:rwxp--aARWcCos:-------:allow
                 group@:rwxp--aARWc--s:-------:allow
```

BEISPIEL 7–13 Interaktion der Zugriffskontrollliste mit chmod-Vorgängen in ZFS-Dateien *(Fortsetzung)*

```
             everyone@:------a-R-c--s:-------:allow
```

Ein chown-Vorgang ändert die Dateieigentümerschaft für `file.1`, und die Ausgabe wird jetzt für den Eigentümer angezeigt, `amy`. Beispiel:

```
# chown amy:staff file.1
# su - amy
$ ls -lV file.1
-rwxrwx---+  1 amy      staff     206695 Aug 30 16:03 file.1
              user:amy:r-----a-R-c---:-------:allow
             user:rory:r-----a-R-c---:-------:allow
         group:sysadmin:rw-p--aARWc---:-------:allow
            group:staff:rw-p--aARWc---:-------:allow
                 owner@:rwxp--aARWcCos:-------:allow
                 group@:rwxp--aARWc--s:-------:allow
             everyone@:------a-R-c--s:-------:allow
```

Der folgende chmod-Vorgang ändert die Zugriffsrechte in einen restriktiveren Modus. In diesem Beispiel überschreiten die geänderten Zugriffsrechte der `sysadmin`- und `staff`-Gruppe die Zugriffsrechte der Eigentümergruppe nicht.

```
$ chmod 640 file.1
$ ls -lV file.1
-rw-r-----+  1 amy      staff     206695 Aug 30 16:03 file.1
              user:amy:r-----a-R-c---:-------:allow
             user:rory:r-----a-R-c---:-------:allow
         group:sysadmin:r-----a-R-c---:-------:allow
            group:staff:r-----a-R-c---:-------:allow
                 owner@:rw-p--aARWcCos:-------:allow
                 group@:r-----a-R-c--s:-------:allow
             everyone@:------a-R-c--s:-------:allow
```

Der folgende chmod-Vorgang ändert die Zugriffsrechte in einen weniger restriktiven Modus. In diesem Beispiel werden die geänderten Zugriffsrechte der `sysadmin`- und `staff`-Gruppe auf die Zugriffsrechte der Eigentümergruppe zurückgesetzt.

```
$ chmod 770 file.1
$ ls -lV file.1
-rwxrwx---+  1 amy      staff     206695 Aug 30 16:03 file.1
              user:amy:r-----a-R-c---:-------:allow
             user:rory:r-----a-R-c---:-------:allow
         group:sysadmin:rw-p--aARWc---:-------:allow
            group:staff:rw-p--aARWc---:-------:allow
                 owner@:rwxp--aARWcCos:-------:allow
                 group@:rwxp--aARWc--s:-------:allow
             everyone@:------a-R-c--s:-------:allow
```

◆ ◆ ◆ KAPITEL 8

Delegierte Oracle Solaris ZFS-Administration

In diesem Kapitel wird erläutert, wie mit der delegierten Administration Benutzern ohne ausreichende Zugriffsrechte ermöglicht werden kann, ZFS-Administrationsaufgaben zu erledigen.

Dieses Kapitel enthält die folgenden Abschnitte:

Delegierte ZFS-Administration im Überblick

Die delegierte ZFS-Administration dient zum Verteilen fein abgestimmter Zugriffsrechte an bestimmte Benutzer, Gruppen oder an "everyone". Es werden zwei Arten der delegierten Zugriffsrechte unterstützt:

- Einzelne Zugriffsrechte wie `create`, `destroy`, `mount`, `snapshot` usw. können ausdrücklich angegeben werden.
- Auch können Gruppen von Zugriffsrechten, so genannte *Zugriffsrechtsätze*, definiert werden. Ein Zugriffsrechtsatz kann zu einem späteren Zeitpunkt aktualisiert werden. Die daraus resultierenden Änderungen wirken sich automatisch auf alle Benutzer des Satzes aus. Zugriffsrechtsätze beginnen mit dem Zeichen `@` und sind auf eine Länge von 64 Zeichen begrenzt. Für die auf das Zeichen `@` folgenden übrigen Zeichen im Namen gelten dieselben Einschränkungen wie für normale ZFS-Dateisystemnamen.

Die delegierte ZFS-Administration bietet ähnliche Möglichkeiten wie das RBAC-Sicherheitsmodell. Die delegierte ZFS-Administration stellt für die Administration von ZFS-Speicher-Pools und -Dateisystemen die folgenden Vorteile dar:

- Bei jeder Migration des ZFS-Speicher-Pools werden seine Zugriffsberechtigungen mit ihm weitergegeben.
- Durch dynamische Vererbung kann gesteuert werden, wie die Zugriffsrechte durch die Dateisysteme weitergegeben werden.
- Möglich ist eine Konfiguration, bei der nur der Ersteller eines Dateisystems dieses wieder löschen kann.
- Zugriffsrechte können gezielt an bestimmte Dateisysteme delegiert werden. Neu erstellte Dateisysteme können Zugriffsrechte automatisch übernehmen.
- Es wird eine einfache NFS-Administration ermöglicht. So kann beispielsweise ein Benutzer mit ausdrücklichen Zugriffsrechten über NFS einen Schnappschuss im entsprechenden `.zfs/snapshot` -Verzeichnis erstellen.

Die delegierte Administration bietet sich für die Verteilung von ZFS-Aufgaben an. Informationen zum Einsatz von RBAC für allgemeine Oracle Solaris-Administrationsaufgaben finden Sie in Teil III, „Roles, Rights Profiles, and Privileges“ in *System Administration Guide: Security Services*.

Deaktivieren von delegierten ZFS-Zugriffsrechten

Die Funktionen der delegierten Administration können durch Setzen der Pool-Eigenschaft `delegation` gesteuert werden. Beispiel:

```
# 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
```

Standardmäßig ist die Eigenschaft `delegation` aktiviert.

Delegieren von ZFS-Zugriffsrechten

Mit dem Befehl `zfs allow` können Sie Zugriffsrechte für ZFS-Dateisysteme an Nicht-Root-Benutzer delegieren. Dafür stehen folgende Möglichkeiten zur Verfügung:

- Einzelne Zugriffsrechte können an einen Benutzer, eine Gruppe oder global delegiert werden.
- Gruppen von Einzelzugriffsrechten können in Form von *Zugriffsrechtsätzen* ebenfalls an Benutzer, Gruppen oder global delegiert werden.
- Zugriffsrechte können dem aktuellen Dateisystem lokal oder allen untergeordneten Dateisystemen des aktuellen Dateisystems delegiert werden.

In der folgenden Tabelle finden Sie eine Beschreibung der delegierbaren Vorgänge und aller abhängigen Zugriffsrechte, die zum Durchführen des delegierten Vorgangs erforderlich sind.

Zugriffsrecht (Unterbefehl)	Beschreibung	Abhängigkeiten
`allow`	Die Berechtigung, eigene Zugriffsrechte an andere Benutzer weiterzugeben.	Das Zugriffsrecht, das gewährt werden soll, muss ebenfalls vorhanden sein.
`clone`	Die Berechtigung, beliebige Schnappschüsse des Datasets zu klonen.	Die Berechtigungen `create` und `mount` müssen im ursprünglichen Dateisystem vorhanden sein.
`create`	Die Berechtigung, untergeordnete Datasets zu erstellen.	Die Berechtigung `mount` muss ebenfalls vorhanden sein.
`destroy`	Die Berechtigung, ein Dataset zu löschen.	Die Berechtigung `mount` muss ebenfalls vorhanden sein.
`diff`	Die Berechtigung, Pfade in einem Dataset zu ermitteln.	Nicht-Root-Benutzer benötigen diese Berechtigung, um den Befehl `zfs diff` zu verwenden.
`hold`	Die Berechtigung, einen Schnappschuss aufzubewahren.	
`mount`	Die Berechtigung, ein Dateisystem ein- und auszuhängen und Geräteverknüpfungen für Volumes zu erstellen oder zu löschen.	
`promote`	Die Berechtigung, einen Klon zu einem Dataset zu machen.	Die Berechtigungen `mount` und `promote` müssen im ursprünglichen Dateisystem vorhanden sein.
`receive`	Die Berechtigung, mit dem Befehl `zfs receive` untergeordnete Dateisysteme zu erstellen.	Die Berechtigungen `mount` und `create` müssen ebenfalls vorhanden sein.
`release`	Die Berechtigung, einen aufbewahrten Schnappschuss freizugeben. Dies kann dazu führen, dass der Schnappschuss gelöscht wird.	
`rename`	Die Berechtigung, einen Dataset umzubenennen.	Die Berechtigungen `create` und `mount` müssen im neuen übergeordneten Objekt vorhanden sein.
`rollback`	Die Berechtigung, einen Schnappschuss wiederherzustellen.	
`send`	Die Berechtigung, einen Schnappschuss-Datenstrom zu senden.	

Zugriffsrecht (Unterbefehl)	Beschreibung	Abhängigkeiten
`share`	Die Berechtigung, ein Dateisystem freizugeben und zu sperren.	`share` und `sharenfs` sind beide erforderlich, um eine NFS-Freigabe zu erstellen. `share` und `sharesmb` sind beide erforderlich, um eine SMB-Freigabe zu erstellen.
`snapshot`	Die Berechtigung, Schnappschüsse von Datasets herzustellen.	

Sie können die folgenden Zugriffsrechte erteilen, wobei diese auf Zugriff, Lesezugriff oder Bearbeitungszugriff beschränkt sein können:

- `groupquota`
- `groupused`
- `userprop`
- `userquota`
- `userused`

Darüber hinaus kann die Administration folgender ZFS-Eigenschaften an Nicht-Root-Benutzer delegiert werden.

- `aclinherit`
- `aclmode`
- `atime`
- `canmount`
- `casesensitivity`
- `checksum`
- `compression`
- `copies`
- `devices`
- `exec`
- `logbias`
- `mountpoint`
- `nbmand`
- `normalization`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`
- `refquota`
- `refreservation`
- `reservation`
- `rstchown`

- secondarycache
- setuid
- sharenfs
- sharesmb
- snapdir
- sync
- utf8only
- version
- volblocksize
- volsize
- vscan
- xattr
- zoned

Einige dieser Eigenschaften können nur bei der Erstellung des Datasets gesetzt werden. Eine Beschreibung dieser Eigenschaften finden Sie unter „ZFS-Eigenschaften“ auf Seite 181.

Delegieren von ZFS-Zugriffsrechten (`zfs allow`)

Die Syntax für `zfs allow` lautet:

`zfs allow -[ldugecs] everyone|`*user*|*group*[,...] *perm*|@*setname*,...] *filesystem*| *volume*

Die folgende `zfs allow`-Syntax (fett gedruckt) gibt an, wem die Zugriffsrechte übertragen werden:

`zfs allow [-uge]|user|group|everyone` `[,...]` *filesystem* | *volume*

Mehrere Entitäten können durch Komma getrennt als Liste angegeben werden. Wenn keine `-uge`-Option angegeben ist, werden die Argumente der Reihenfolge nach als das Schlüsselwort `everyone`, dann als Benutzername und schließlich als Gruppenname interpretiert. Um einen Benutzer oder eine Gruppe mit dem Namen "everyone " anzugeben, verwenden Sie die Option `-u` bzw. `-g`. Mit der Option `-g` geben Sie eine Gruppe mit demselben Namen eines Benutzers an. Die Option `-c` delegiert "Create-time"-Zugriffsrechte.

Die folgende `zfs allow`-Syntax (fett gedruckt) veranschaulicht, wie Zugriffsrechte und Zugriffsrechtsätze angegeben werden:

`zfs allow [-s] ... perm|@setname [,...]` *filesystem* | *volume*

Mehrere Zugriffsrechte können durch Komma getrennt als Liste angegeben werden. Die Namen der Zugriffsrechte sind mit den ZFS-Unterbefehlen und -Eigenschaften identisch. Weitere Informationen finden Sie im vorherigen Abschnitt.

Mehrere Zugriffsrechte lassen sich in *Zugriffsrechtsätzen* zusammenfassen. Sie werden durch die Option `-s` gekennzeichnet. Zugriffsrechtsätze können von anderen `zfs allow`-Befehlen für das angegebene Dateisystem und dessen untergeordnete Objekte verwendet werden.

Zugriffsrechtsätze werden dynamisch ausgewertet, sodass Änderungen an einem Satz unverzüglich aktualisiert werden. Für Zugriffsrechtsätze gelten dieselben Benennungsanforderungen wie für ZFS-Dateisysteme, wobei der Name allerdings mit dem Zeichen `@` beginnen muss und nicht mehr als 64 Zeichen lang sein darf.

Die folgende `zfs allow`-Syntax (fett gedruckt) gibt an, wie die Zugriffsrechte übertragen werden:

`zfs allow [-ld]` `...` `...` *filesystem* | *volume*

Die Option `-l` gibt an, dass die Zugriffsrechte für das angegebene Dateisystem, nicht aber für dessen untergeordnete Objekte gewährt werden, es sei denn, es wurde auch die Option `-d` angegeben. Die Option `-d` bedeutet, dass die Zugriffsrechte nicht für dieses Dateisystem, sondern für dessen untergeordnetes Dateisysteme gewährt werden, es sei denn, es wurde auch die Option `-l` angegeben. Wenn keine der Optionen angegeben wird, gelten die Zugriffsrechte für das Dateisystem bzw. Volume und alle untergeordneten Objekte.

Löschen von delegierten ZFS-Zugriffsrechten (`zfs unallow`)

Mit dem Befehl `zfs unallow` können Sie zuvor delegierte Zugriffsrechte wieder löschen.

Gehen wir beispielsweise davon aus, dass Sie die Zugriffsrechte `create`, `destroy`, `mount` und `snapshot` wie folgt delegiert haben:

```
# zfs allow cindy create,destroy,mount,snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy --------------------------------
Local+Descendent permissions:
        user cindy create,destroy,mount,snapshot
```

Zum Entfernen dieser Zugriffsrechte verwenden Sie folgende Syntax:

```
# zfs unallow cindy tank/home/cindy
# zfs allow tank/home/cindy
```

Delegieren von ZFS-Zugriffsrechten (Beispiele)

BEISPIEL 8–1 Delegieren von Zugriffsrechten an einzelne Benutzer

Wenn Sie die Zugriffsrechte `create` und `mount` an einen einzelnen Benutzer delegieren, müssen Sie sich vergewissern, dass dieser über die erforderlichen Zugriffsrechte für den zugrunde liegenden Einhängepunkt verfügt.

Um beispielsweise an den Benutzer `mark` die Zugriffsrechte `create` und `mount` für das Dateisystem `tank` zu delegieren, legen Sie zuerst die Zugriffsrechte fest:

BEISPIEL 8–1 Delegieren von Zugriffsrechten an einzelne Benutzer *(Fortsetzung)*

```
# chmod A+user:mark:add_subdirectory:fd:allow /tank/home
```

Anschließend delegieren Sie mit dem Befehl `zfs allow` die Zugriffsrechte `create`, `destroy` und `mount`. Beispiel:

```
# zfs allow mark create,destroy,mount tank/home
```

Jetzt kann der Benutzer `mark` im Dateisystem `tankHome` eigene Dateisysteme anlegen. Beispiel:

```
# su mark
mark$ zfs create tank/home/mark
mark$ ^D
# su lp
$ zfs create tank/home/lp
cannot create 'tank/home/lp': permission denied
```

BEISPIEL 8–2 Delegieren der Zugriffsrechte create und destroy an Gruppen

Das folgende Beispiel veranschaulicht, wie ein Dateisystem eingerichtet wird, damit jedes Mitglied der Gruppe `staff` Dateisysteme unter `tank/home` erstellen und einhängen sowie eigene Dateisysteme löschen kann. Die Gruppenmitglieder von `staff` können jedoch nicht die Dateisysteme anderer Gruppenmitglieder löschen.

```
# zfs allow staff create,mount tank/home
# zfs allow -c create,destroy tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
        create,destroy
Local+Descendent permissions:
        group staff create,mount
# su cindy
cindy% zfs create tank/home/cindy/files
cindy% exit
# su mark
mark% zfs create tank/home/mark/data
mark% exit
cindy% zfs destroy tank/home/mark/data
cannot destroy 'tank/home/mark/data': permission denied
```

BEISPIEL 8–3 Delegieren von Zugriffsrechten auf der richtigen Ebene der Dateisystemhierarchie

Stellen Sie sicher, dass Sie auf der richtigen Ebene der Dateisystemhierarchie die Zugriffsrechte an die Benutzer delegieren. So werden beispielsweise an den Benutzer `mark` die Zugriffsrechte `create`, `destroy` und `mount` für das lokale und für die untergeordneten Dateisysteme delegiert. An den Benutzer `mark` werden lokale Zugriffsrechte zum Erstellen von Schnappschüssen des Dateisystems `tank/home`, nicht aber des eigenen Dateisystems, delegiert. Das Zugriffsrecht `snapshot` wurde also nicht auf der richtigen Ebene der Dateisystemhierarchie an ihn delegiert.

BEISPIEL 8–3 Delegieren von Zugriffsrechten auf der richtigen Ebene der Dateisystemhierarchie *(Fortsetzung)*

```
# zfs allow -l mark snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
        create,destroy
Local permissions:
        user mark snapshot
Local+Descendent permissions:
        group staff create,mount
# su mark
mark$ zfs snapshot tank/home@snap1
mark$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

Um an den Benutzer `mark` Zugriffsrechte für die untergeordnete Ebene zu delegieren, verwenden Sie den Befehl `zfs allow` mit der Option `-d`. Beispiel:

```
# zfs unallow -l mark snapshot tank/home
# zfs allow -d mark snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
        create,destroy
Descendent permissions:
        user mark snapshot
Local+Descendent permissions:
        group staff create,mount
# su mark
$ zfs snapshot tank/home@snap2
cannot create snapshot 'tank/home@snap2': permission denied
$ zfs snapshot tank/home/mark@snappy
```

Jetzt kann der Benutzer `mark` nur Schnappschüsse unterhalb der Dateisystemebene `tank/home` erstellen.

BEISPIEL 8–4 Definieren und Anwenden komplexer delegierter Zugriffsrechte

Es können spezifische Zugriffsrechte an Benutzer oder Gruppen delegiert werden. So werden beispielsweise mit dem folgenden `zfs allow`-Befehl spezifische Zugriffsrechte an die Gruppe `staff` delegiert. Darüber hinaus werden die Zugriffsrechte `destroy` und `snapshot` delegiert, sobald `tank/home`-Dateisysteme erstellt werden.

```
# zfs allow staff create,mount tank/home
# zfs allow -c destroy,snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
        create,destroy,snapshot
Local+Descendent permissions:
        group staff create,mount
```

BEISPIEL 8–4 Definieren und Anwenden komplexer delegierter Zugriffsrechte *(Fortsetzung)*

Da Benutzer `mark` ein Mitglied der Gruppe `staff` ist, kann er in `tank/home` Dateisysteme erstellen. Zusätzlich kann Benutzer `mark` Schnappschüsse von `tank/home/mark2` erstellen, da er über die dafür erforderlichen Zugriffsrechte verfügt. Beispiel:

```
# su mark
$ zfs create tank/home/mark2
$ zfs allow tank/home/mark2
---- Permissions on tank/home/mark2 --------------------------------
Local permissions:
        user mark create,destroy,snapshot
---- Permissions on tank/home --------------------------------------
Create time permissions:
        create,destroy,snapshot
Local+Descendent permissions:
        group staff create,mount
```

Benutzer `mark` kann jedoch keine Schnappschüsse in `tank/home/mark` erstellen, da ihm die Zugriffsrechte hierfür fehlen. Beispiel:

```
$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

In diesem Beispiel verfügt der Benutzer `mark` über das Zugriffsrecht `create` in seinem Home-Verzeichnis. Er kann also Schnappschüsse erstellen. Dies kann sich bei Dateisystemen als hilfreich erweisen, die per NFS eingehängt werden.

```
$ cd /tank/home/mark2
$ ls
$ cd .zfs
$ ls
shares snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x   2 mark    staff          2 Sep 27 15:55 snap1
$ pwd
/tank/home/mark2/.zfs/snapshot
$ mkdir snap2
$ zfs list
# zfs list -r tank/home
NAME                    USED  AVAIL  REFER  MOUNTPOINT
tank/home/mark           63K  62.3G    32K  /tank/home/mark
tank/home/mark2          49K  62.3G    31K  /tank/home/mark2
tank/home/mark2@snap1    18K      -    31K  -
tank/home/mark2@snap2      0      -    31K  -
$ ls
snap1  snap2
$ rmdir snap2
$ ls
snap1
```

BEISPIEL 8–5 Definieren und Anwenden von delegierten Zugriffsrechtsätzen für ZFS

Das folgende Beispiel zeigt, wie der Zugriffsrechtsatz `@myset` erstellt wird und dieser sowie das Zugriffsrecht `rename` an die Gruppe `staff` für das Dateisystem `tank` delegiert werden. Der Benutzer `cindy`, Mitglied der Gruppe `staff`, verfügt über die Berechtigung zum Erstellen von Dateisystemen in `tank`. Benutzer `lp` verfügt jedoch nicht über die Berechtigung zum Erstellen von Dateisystemen in `tank`.

```
# zfs allow -s @myset create,destroy,mount,snapshot,promote,clone,readonly tank
# zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
        @myset clone,create,destroy,mount,promote,readonly,snapshot
# zfs allow staff @myset,rename tank
# zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
        @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions:
        group staff @myset,rename
# chmod A+group:staff:add_subdirectory:fd:allow tank
# su cindy
cindy% zfs create tank/data
cindy% zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
        @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions:
        group staff @myset,rename
cindy% ls -l /tank
total 15
drwxr-xr-x   2 cindy   staff          2 Jun 24 10:55 data
cindy% exit
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied
```

Anzeigen von delegierten ZFS-Zugriffsrechten (Beispiele)

Mit dem folgenden Befehl werden die Zugriffsrechte angezeigt:

```
# zfs allow dataset
```

Dieser Befehl gibt die für das angegebene Dataset gesetzten oder gewährten Zugriffsrechte aus. Die Ausgabe enthält folgende Komponenten:

- Zugriffsrechtsätze
- Individuelle Zugriffsrechte oder Create-time-Zugriffsrechte
- Lokales Dataset
- Lokale und untergeordnete Datasets
- Nur untergeordnete Datasets

BEISPIEL 8–6 Anzeigen einfacher Zugriffsrechte für die delegierte Administration

Die folgende Ausgabe besagt, dass Benutzer `cindy` über die Zugriffsrechte `create`, `destroy`, `mount` und "snapshot" im Dateisystem `tank/cindy` verfügt.

```
# zfs allow tank/cindy
       -------------------------------------------------------------
       Local+Descendent permissions on (tank/cindy)
               user cindy create,destroy,mount,snapshot
```

BEISPIEL 8–7 Anzeigen komplexer Zugriffsrechte für die delegierte Administration

Die Ausgabe in diesem Beispiel zeigt die folgenden Zugriffsrechte für die Dateisysteme `pool/fred` und `pool` auf.

Für das Dateisystem `pool/fred`:

- Es sind zwei Zugriffsrechtsätze definiert:
  - `@eng` (`create`, `destroy` , `snapshot`, `mount`, `clone` , `promote`, `rename`)
  - `@simple` (`create`, `mount`)
- Die Create-time-Zugriffsrechte werden für den Zugriffsrechtsatz `@eng` und die Eigenschaft `mountpoint` gesetzt. "Create-time" bedeutet, dass der Zugriffsrechtsatz `@eng` und die Berechtigung zum Festlegen der Eigenschaft `mountpoint` nach der Erstellung eines Dateisystems delegiert werden.
- An den Benutzer `tom` wird der Zugriffsrechtsatz `@eng` und an den Benutzer `joe` werden die Zugriffsrechte `create`, `destroy` und `mount` für lokale Dateisysteme delegiert.
- An den Benutzer `fred` werden der Zugriffsrechtsatz `@basic` sowie die Zugriffsrechte `share` und `rename` für das lokale und die untergeordneten Dateisysteme delegiert.
- An den Benutzer `barney` und der Gruppe `staff` wird der Zugriffsrechtsatz `@basic` ausschließlich für untergeordnete Dateisysteme delegiert.

Für das Dateisystem `pool`:

- Der Zugriffsrechtsatz `@simple` (create, destroy, mount) ist definiert.
- Die Gruppe `staff` erhält den Zugriffsrechtsatz `@simple` für das lokale Dateisystem.

Sehen Sie hier die Ausgabe für dieses Beispiel:

```
$ zfs allow pool/fred
---- Permissions on pool/fred --------------------------------------
Permission sets:
        @eng create,destroy,snapshot,mount,clone,promote,rename
        @simple create,mount
Create time permissions:
        @eng,mountpoint
Local permissions:
        user tom @eng
        user joe create,destroy,mount
Local+Descendent permissions:
```

BEISPIEL 8–7 Anzeigen komplexer Zugriffsrechte für die delegierte Administration *(Fortsetzung)*

```
        user fred @basic,share,rename
        user barney @basic
        group staff @basic
---- Permissions on pool --------------------------------------------
Permission sets:
        @simple create,destroy,mount
Local permissions:
        group staff @simple
```

Löschen von delegierten ZFS-Zugriffsrechten (Beispiele)

Mit dem Befehl `zfs unallow` lassen sich delegierte Zugriffsrechte wieder löschen. Beispielsweise verfügt Benutzer `cindy` über die Zugriffsrechte `create`, `destroy`, `mount` und `snapshot` im Dateisystem `tank/cindy`.

```
# zfs allow cindy create,destroy,mount,snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy ---------------------------------
Local+Descendent permissions:
        user cindy create,destroy,mount,snapshot
```

Mit der folgenden `zfs unallow`-Syntax entziehen Sie das Zugriffsrecht `snapshot` von Benutzer `cindy` für das Dateisystem `tank/home/cindy`:

```
# zfs unallow cindy snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy ---------------------------------
Local+Descendent permissions:
        user cindy create,destroy,mount
cindy% zfs create tank/home/cindy/data
cindy% zfs snapshot tank/home/cindy@today
cannot create snapshot 'tank/home/cindy@today': permission denied
```

Nehmen wir in einem anderen Beispiel an, dass Benutzer `mark` über folgende Zugriffsrechte im Dateisystem `tank/home/mark` verfügt:

```
# zfs allow tank/home/mark
---- Permissions on tank/home/mark ----------------------------------
Local+Descendent permissions:
        user mark create,destroy,mount
-------------------------------------------------------------
```

Mit der folgenden `zfs unallow`-Syntax entziehen Sie Benutzer `mark` alle Zugriffsrechte für das Dateisystem `tank/home/mark`:

```
# zfs unallow mark tank/home/mark
```

Die folgende `zfs unallow`-Syntax löscht einen Zugriffsrechtsatz für das Dateisystem `tank`.

```
# zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
        @myset clone,create,destroy,mount,promote,readonly,snapshot
Create time permissions:
        create,destroy,mount
Local+Descendent permissions:
        group staff create,mount
# zfs unallow -s @myset tank
# zfs allow tank
---- Permissions on tank --------------------------------------------
Create time permissions:
        create,destroy,mount
Local+Descendent permissions:
        group staff create,mount
```

◆ ◆ ◆ KAPITEL 9

Fortgeschrittene Oracle Solaris ZFS-Themen

Dieses Kapitel enthält Informationen zu ZFS-Volumes, zur Verwendung von ZFS auf Solaris-Systemen mit installierten Zonen, zu Speicher-Pools mit alternativem Root-Verzeichnis sowie zu ZFS-Zugriffsrechtsprofilen.

Dieses Kapitel enthält die folgenden Abschnitte:

ZFS-Volumes

Ein ZFS-Volume ist ein Dataset, das eine Blockeinheit darstellt. ZFS-Volumes werden im Verzeichnis `/dev/zvol/{dsk,rdsk}/pool` als Geräte identifiziert.

Im folgenden Beispiel wird ein ZFS-Volume `tank/vol` mit einer Kapazität von 5 GB erstellt:

```
# zfs create -V 5gb tank/vol
```

Beim Erstellen von Volumes wird automatisch eine Reservierung der anfänglichen Volume-Kapazität angelegt, um unerwartetes Verhalten zu verhindern. Wenn die Kapazität des Volumes beispielsweise kleiner wird, können Daten beschädigt werden. Deswegen müssen Sie beim Ändern der Kapazität eines Volumes äußerst sorgfältig vorgehen.

Außerdem können Dateisysteminkonsistenzen entstehen, wenn Sie einen Schnappschuss eines Volumes erstellen, dessen Kapazität sich ändern kann, und versuchen, den betreffenden Schnappschuss mittels Rollback rückgängig zu machen oder zu klonen.

Informationen zu Dateisystemeigenschaften, die auf Volumes angewendet werden können, finden Sie in Tabelle 5–1.

Sie können die Eigenschaftsinformationen eines ZFS-Volumes mit dem Befehl `zfs get` oder `zfs get all` anzeigen. Beispiel:

```
# zfs get all tank/vol
```

Ein Fragezeichen (?), das für `volsize` in der `zfs get`-Ausgabe angezeigt wird, weist auf einen unbekannten Wert wegen eines E/A-Fehlers hin. Beispiel:

```
# zfs get -H volsize tank/vol
tank/vol        volsize ?       local
```

Ein E7A-Fehler weist im Allgemeinen auf ein Problem mit einem Poolgerät hin. Weitere Informationen über die Abhilfe bei Problemen mit Poolgeräten finden Sie in „Identifizieren von Problemen mit ZFS-Speicherpools“ auf Seite 298.

Auf Solaris-Systemen mit installierten Zonen können Sie keine ZFS-Volumes in einer nicht globalen Zone erstellen bzw. klonen. Alle solche Versuche schlagen fehl. Informationen zur Verwendung von ZFS-Volumes in einer globalen Zone finden Sie unter „Hinzufügen von ZFS-Volumes zu einer nicht globalen Zone“ auf Seite 289.

Verwendung von ZFS-Volumes als Swap- bzw. Dump-Gerät

Während der Installation eines ZFS-Root-Dateisystems oder einer Migration von einem UFS-Root-Dateisystem wird auf einem ZFS-Volume im ZFS-Root-Pool ein Swap-Gerät erstellt. Beispiel:

```
# swap -l
swapfile                  dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap 253,3        16  8257520  8257520
```

Während der Installation eines ZFS-Root-Dateisystems oder einer Migration von einem UFS-Root-Dateisystem wird auf einem ZFS-Volume im ZFS-Root-Pool ein Dump-Gerät erstellt. Nach der Einrichtung ist keine Administration des Dump-Geräts erforderlich. Beispiel:

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/
  Savecore enabled: yes
```

Wenn Sie den Swap-Bereich oder das Dump-Gerät nach der Installation des Systems ändern müssen, benutzen Sie hierzu die Befehle `swap` und `dumpadm` wie in vorherigen Solaris-Versionen. Zum Erstellen eines zusätzlichen Swap-Volumes müssen Sie ein ZFS-Volume einer bestimmten Kapazität erstellen und für dieses Gerät dann die Swap-Funktion aktivieren. Fügen Sie dann einen Eintrag für das neue Swap-Gerät in die Datei `/etc/vfstab` ein. Beispiel:

```
# 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
```

Swaps dürfen nicht in eine Datei eines ZFS-Dateisystems durchgeführt werden. ZFS unterstützt keine Konfigurationen für Swap-Dateien.

Informationen zum Anpassen der Größe von Swap- und Dump-Volumes finden Sie unter „Anpassen der Größe von ZFS-Swap- und Dump-Geräten“ auf Seite 159.

Verwendung von ZFS-Volumes als Solaris-iSCSI-Zielgerät

Sie können ZFS-Volumes auf einfache Weise als iSCSI-Zielgerät konfigurieren, indem Sie die Eigenschaft `shareiscsi` für das Volume setzen. Beispiel:

```
# 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
```

Nach dem Erstellen des iSCSI-Zielgeräts muss der iSCSI-Initiator definiert werden. Weitere Informationen zu Solaris-iSCSI-Zielgeräten und -Initiatoren finden Sie in Kapitel 12, „Configuring Oracle Solaris iSCSI Targets (Tasks)“ in *System Administration Guide: Devices and File Systems*.

Hinweis – Solaris-iSCSI-Zielgeräte können auch mit dem Befehl `iscsitadm` erstellt und verwaltet werden. Wenn die Eigenschaft `shareiscsi` eines ZFS-Volumes gesetzt ist, darf der Befehl `iscsitadm` nicht zum Erstellen des gleichen Zielgeräts verwendet werden, da ansonsten für das gleiche Zielgerät duplizierte Zielgerätinformationen erstellt werden.

Als iSCSI-Zielgerät konfigurierte ZFS-Volumes werden genauso wie andere ZFS-Datasets verwaltet. Operationen zum `Umbenennen`, `Exportieren` und `Importieren` funktionieren bei iSCSI-Zielgeräten jedoch etwas anders.

- Wenn Sie ein ZFS-Volume umbenennen, bleibt der Name des iSCSI-Zielgeräts gleich. Beispiel:

  ```
  # 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
  ```

- Durch das Exportieren eines Pools, das ein für den Netzwerkzugriff freigegebenes ZFS-Volume enthält, wird das betreffende Zielgerät entfernt. Durch das Importieren eines Pools, das ein freigegebenes ZFS-Volume enthält, wird das betreffende Zielgerät für den Netzwerkzugriff freigegeben. Beispiel:

```
# 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
```

Alle Konfigurationsinformationen zu einem iSCSI-Zielgerät werden im Dataset gespeichert. Wie bei einem über NFS für den Netzwerkzugriff freigegebenem Dateisystem, wird ein in ein anderes System importiertes iSCSI-Zielgerät entsprechend für den Netzwerkzugriff freigegeben.

Verwendung von ZFS in einem Solaris-System mit installierten Zonen

In den folgenden Abschnitten wird die Verwendung von ZFS auf Systemen mit Oracle Solaris-Zonen beschrieben:

Informationen zum Konfigurieren eines Systems mit einem ZFS-Root-Dateisystem, das mithilfe von Live Upgrade migriert oder gepatcht werden soll, finden Sie unter „Verwenden von Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (Solaris 10 10/08)“ auf Seite 142 oder „Verwenden des Oracle Solaris Live Upgrade zum Migrieren oder Aktualisieren eines Systems mit Zonen (ab Solaris 10 5/09)“ auf Seite 147.

Für die Zusammenarbeit von ZFS-Datasets mit Zonen sollten Sie folgende Aspekte berücksichtigen:

- Sie können ein ZFS-Dateisystem bzw. einen ZFS-Klon mit oder ohne Delegierung der Administrationssteuerung zu einer nicht globalen Zone hinzufügen.
- ZFS-Volumes können als Gerät zu nicht globalen Zonen hinzugefügt werden.
- Derzeit können ZFS-Schnappschüsse nicht in Zonen erstellt werden.

In den folgenden Abschnitten wird mit dem Begriff "ZFS-Dataset" ein Dateisystem bzw. Klon bezeichnet.

Durch Hinzufügen eines Datasets kann sich eine nicht globale Zone Festplattenkapazität mit der globalen Zone teilen, obwohl der Zonen-Administrator in der zugrunde liegenden Dateisystemhierarchie keine Eigenschaften einstellen bzw. keine neuen Dateisysteme erstellen kann. Dieser Vorgang ist mit dem Hinzufügen anderer Dateisystemtypen zu einer Zone identisch und sollte dann verwendet werden, wenn das Hauptziel in der gemeinsamen Nutzung von Festplattenkapazität besteht.

In ZFS können Datasets auch an eine nicht globale Zone delegiert werden. Dadurch erlangt der Zonen-Administrator vollständige Kontrolle über diese Datasets und alle seine untergeordneten Objekte. Der Zonen-Administrator kann Dateisysteme bzw. Klone in diesem Dataset erstellen oder löschen sowie Dataset-Eigenschaften ändern. Der Zonen-Administrator hat jedoch keine Kontrolle über Datasets, die nicht zur betreffenden Zone hinzugefügt wurden, und darf die für die oberste Hierarchieebene des delegierten Datasets gesetzten Kontingente nicht überschreiten.

Bei der Verwendung von ZFS auf Systemen mit installierten Oracle Solaris-Zonen sollten Sie Folgendes berücksichtigen:

- Die Eigenschaft `mountpoint eines zu einer nicht globalen Zone hinzugefügten ZFS-Dateisystems muss auf "legacy" gesetzt sein`.
- Sehen Sie davon ab, einer nicht globalen Zone bei der Konfiguration ein ZFS-Dataset hinzuzufügen. Sie können das•ZFS-Dataset hinzufügen, nachdem die Zone installiert wurde.
- Befinden sich der `zonepath` der Quelle und der `zonepath` des Ziels in einem ZFS-Dateisystem und im gleichen Pool, so verwendet der Befehl `zoneadm clone` nun automatisch die ZFS-Klonfunktion, um die Zone zu klonen. Mit dem Befehl `zoneadm clone` wird ein ZFS-Schnappschuss des `zonepath`-Quellverzeichnisses erstellt und das `zonepath`-Zielverzeichnis eingerichtet. Das Klonen einer Zone mit dem Befehl `zfs clone` ist nicht möglich. Weitere Informationen finden Sie in Teil II, „Zonen“ in *SystemAdministrationshandbuch: Oracle Solaris Container – RessourcenAdministration und Solaris Zones*.
- Wenn Sie ein ZFS-Dateisystem an eine nicht globale Zone delegieren, müssen Sie dieses Dateisystem vor der Verwendung des Oracle Solaris Live Upgrade aus der nicht globalen Zone entfernen. Andernfalls kann das Oracle Live Upgrade nicht ausgeführt werden, was auf einen Fehler im Zusammenhang mit dem Schreibschutz des Dateisystems zurückzuführen ist.

Hinzufügen von ZFS-Dateisystemen zu einer nicht globalen Zone

Wenn das Hauptziel ausschließlich in der gemeinsamen Nutzung von Speicherplatz mit der globalen Zone besteht, können Sie ein ZFS-Dateisystem zu einer nicht globalen Zone als generisches Dateisystem hinzufügen. Die Eigenschaft `mountpoint eines zu einer nicht`

globalen Zone hinzugefügten ZFS-Dateisystems muss auf "legacy" gesetzt sein. Stellen Sie beispielsweise beim Hinzufügen des Dateisystems tank/zone/zion zu einer nicht globalen Zone die Eigenschaft mountpoint in der globalen Zone wie folgt ein:

```
# zfs set mountpoint=legacy tank/zone/zion
```

Sie können ein ZFS-Dateisystem mit dem Unterbefehl add fs des Befehls zonecfg zu einer nicht globalen Zone hinzufügen.

Im folgenden Beispiel wird ein ZFS-Dateisystem durch den Administrator der globalen Zone aus der globalen Zone zu einer nicht globalen Zone hinzugefügt:

```
# 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=/opt/data
zonecfg:zion:fs> end
```

Diese Syntax fügt das ZFS-Dateisystem tank/zone/zion zur bereits konfigurierten und unter /opt/data eingehängten Zone zion hinzu. Die Eigenschaft mountpoint des Dateisystems muss auf legacy gesetzt sein, und das Dateisystem darf nicht schon irgendwo anders eingehängt sein. Der Zonenadministrator kann Dateien im Dateisystem erstellen und löschen. Das Dateisystem kann nicht an einer anderen Stelle eingehängt werden, und der Zonenadministrator kann keine Dateisystemeigenschaften wie z. B. atime, readonly, compression usw. ändern. Der Administrator der globalen Zone ist für das Einstellen und Steuern von Dateisystemeigenschaften verantwortlich.

Weitere Informationen zum Befehl zonecfg sowie zur Konfiguration von Ressourcentypen mithilfe von zonecfg finden Sie unter Teil II, „Zonen“ in *SystemAdministrationshandbuch: Oracle Solaris Container – RessourcenAdministration und Solaris Zones*.

Delegieren von Datasets in eine nicht globale Zone

Wenn das Hauptziel in der Delegierung der Speicherplatzadministration an eine Zone besteht, können Datasets mit dem Unterbefehl add dataset des Befehls zonecfg zu einer nicht globalen Zone hinzugefügt werden.

Im folgenden Beispiel wird ein ZFS-Dateisystem durch den Administrator der globalen Zone aus der globalen Zone an eine nicht globale Zone delegiert:

```
# zonecfg -z zion
zonecfg:zion> add dataset
zonecfg:zion:dataset> set name=tank/zone/zion
zonecfg:zion:dataset> end
```

Im Gegensatz zum Hinzufügen von Dateisystemen wird durch diese Syntax das ZFS-Dateisystem tank/zone/zion in der bereits konfigurierten Zone zion sichtbar. Der

Zonenadministrator kann Dateisystemeigenschaften einstellen und untergeordnete Dateisysteme erstellen. Darüber hinaus kann der Zonenadministrator Schnappschüsse sowie Klone erstellen und die Dateisystemhierarchie auch anderweitig steuern.

Im Gegensatz zum Hinzufügen von Dateisystemen wird durch diese Syntax das ZFS-Dateisystem `tank/zone/zion` in der bereits konfigurierten Zone `zion` sichtbar. Innerhalb der `zion`-Zone kann auf dieses Dateisystem nicht als `tank/zone/zion` zugegriffen werden, sondern als *virtueller Pool* namens `tank`. Der Alias des delegierten Dateisystems stellt eine Ansicht des Originalpools als virtueller Pool in der Zone bereit. Die Aliaseigenschaft gibt den Namen des virtuellen Pools an. Wenn kein Alias angegeben wird, wird ein Standardalias, der mit der letzten Komponente des Dateisystemnamens übereinstimmt, verwendet. Wenn kein spezifischer Alias angegeben wird, wäre der Standardalias im obigen Beispiel `zion` gewesen.

Bei delegierten Datasets kann der Zonenadministrator Dateisystemeigenschaften einstellen und untergeordnete Dateisysteme erstellen. Darüber hinaus kann der Zonenadministrator Schnappschüsse sowie Klone erstellen und die Dateisystemhierarchie auch anderweitig steuern. Wenn ZFS-Volumes innerhalb von delegierten Dateisystemen erstellt werden, sind diese möglicherweise nicht mit ZFS-Volumes vereinbar, die als Geräteressourcen hinzugefügt werden. Weitere Informationen finden Sie im folgenden Abschnitt.

Wenn Sie Oracle Solaris Live Upgrade zur Aktualisierung der ZFS-BU mit nicht globalen Zonen verwenden, sollten Sie vorher jegliche delegierte Datasets entfernen. Andernfalls kann das Oracle Solaris Live Upgrade nicht ausgeführt werden, was auf einen Fehler im Zusammenhang mit dem Schreibschutz des Dateisystems zurückzuführen ist. Beispiel:

```
zonecfg:zion>
zonecfg:zion> remove dataset name=tank/zone/zion
zonecfg:zion1> exit
```

Weitere Informationen zu zulässigen Aktionen innerhalb von Zonen finden Sie unter „Verwalten von ZFS-Eigenschaften innerhalb einer Zone“ auf Seite 290.

Hinzufügen von ZFS-Volumes zu einer nicht globalen Zone

ZFS-Volumes können nicht mit dem Unterbefehl `add dataset` des Befehls `zonecfg` zu einer nicht globalen Zone hinzugefügt werden. Mithilfe des Unterbefehls `add device` des Befehls `zonecfg` können Zonen jedoch Volumes hinzugefügt werden.

Im folgenden Beispiel wird ein ZFS-Volume durch den Administrator der globalen Zone aus der globalen Zone zu einer nicht globalen Zone hinzugefügt:

```
# 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
```

Mit dieser Syntax wird das Volume `tank/vol` zur Zone`zion` hinzugefügt.

Bitte beachten Sie, dass das Hinzufügen eines Volumes im Raw-Modus auch dann Sicherheitsrisiken in sich birgt, wenn das Volume keinem physischen Datenspeichergerät entspricht. Der Zonenadministrator könnte dadurch insbesondere fehlerhafte Dateisysteme erstellen, die beim Einhängen einen Systemabsturz verursachen. Weitere Informationen zum Hinzufügen von Geräten zu Zonen und den damit verbundenen Sicherheitsrisiken finden Sie unter „Informationen zur Eigenschaft `zoned`“ auf Seite 291.

Weitere Informationen zum Hinzufügen von Geräten zu Zonen finden Sie unter Teil II, „Zonen“ in *SystemAdministrationshandbuch: Oracle Solaris Container – RessourcenAdministration und Solaris Zones*.

Verwenden von ZFS-Speicher-Pools innerhalb einer Zone

ZFS-Speicher-Pools können in Zonen werder erstellt noch geändert werden. Das Modell der delegierten Administration zentralisiert die Administration physischer Datenspeichergeräte in der globalen Zone und die Administration von virtuellem Speicherplatz in den nicht globalen Zonen. Obwohl Datasets auf Pool-Ebene zu einer Zone hinzugefügt werden können, sind Befehle, die die physischen Eigenschaften eines Pools ändern (z. B. Erstellen, Hinzufügen oder Entfernen von Datenspeichergeräten) innerhalb einer Zone nicht zulässig. Auch wenn physische Datenspeichergeräte mit dem Unterbefehl `add device` des Befehls `zonecfg` zu einer Zone hinzugefügt oder Dateien verwendet werden, erlaubt der Befehl `zpool` kein Erstellen neuer Pools innerhalb der Zone.

Verwalten von ZFS-Eigenschaften innerhalb einer Zone

Nach dem Delegieren eines Datasets an eine Zone kann der Zonenadministrator bestimmte Dataset-Eigenschaften einstellen. Wenn ein Dataset an eine Zone delegiert wird, sind alle seine übergeordneten Datasets als schreibgeschützte Datasets sichtbar. Das Dataset selbst sowie alle seine untergeordneten Datasets besitzen diesen Schreibschutz jedoch nicht. Betrachten wir zum Beispiel die folgende Konfiguration:

```
global# zfs list -Ho name
tank
tank/home
```

```
tank/data
tank/data/matrix
tank/data/zion
tank/data/zion/home
```

Würde `tank/data/zion` zu einer Zone hinzugefügt, besäße jedes Dataset folgende Eigenschaften.

Dataset	Sichtbar	Beschreibbar	Unveränderbare Eigenschaften
`tank`	Ja	Nein	-
`tank/home`	Nein	-	-
`tank/data`	Ja	Nein	-
`tank/data/matrix`	Nein	-	-
`tank/data/zion`	Ja	Ja	`sharenfs`, `zoned`, `quota`, `reservation`
`tank/data/zion/home`	Ja	Ja	`sharenfs`, `zoned`

Beachten Sie, dass übergeordnete Datasets von `tank/zone/zion` schreibgeschützt sichtbar, alle untergeordneten Datasets beschreibbar und nicht zur übergeordneten Hierarchie gehörende Datasets nicht sichtbar sind. Der Zonenadministrator kann die Eigenschaft `sharenfs` nicht ändern, da nicht globale Zonen nicht als NFS-Server fungieren können. Der Zonenadministrator kann die Eigenschaft `zoned` nicht ändern, da dies ein Sicherheitsrisiko (siehe nächster Abschnitt) darstellen würde.

Privilegierte Benutzer in der Zone können jede andere einstellbare Eigenschaft ändern, ausgenommen die Eigenschaften `quota` und `reservation`. Durch dieses Verhalten kann der globale Zonenadministrator den Verbrauch von Festplattenkapazität aller Datasets in den nicht globalen Zonen überwachen.

Außerdem können nach dem Delegieren eines Datasets an eine nicht globale Zone die Eigenschaften `sharenfs` und `mountpoint` vom globalen Zonenadministrator nicht geändert werden.

Informationen zur Eigenschaft `zoned`

Beim Delegieren eines Datasets an eine nicht globale Zone muss dieses Dataset entsprechend gekennzeichnet werden, sodass bestimmte Eigenschaften nicht im Kontext der globalen Zone interpretiert werden. Nachdem ein Dataset an eine nicht globale Zone delegiert wurde und unter der Kontrolle eines Zonenadministrators ist, ist dessen Inhalt nicht mehr vertrauenswürdig. Wie bei anderen Dateisystemen auch können `setuid`-Binärdateien, symbolische Verknüpfungen oder andere fragwürdige Inhalte auftreten, die sich negativ auf die

Sicherheit der globalen Zone auswirken. Darüber hinaus kann die Eigenschaft `mountpoint` nicht im Kontext der globalen Zone interpretiert werden, da der Zonenadministrator andernfalls den Namespace der globalen Zone ändern kann. Zur Lösung des letzteren Problems nutzt ZFS die Eigenschaft `zoned`, die anzeigt, dass ein Dataset zu einem bestimmten Zeitpunkt an eine nicht globale Zone delegiert wurde.

Die Eigenschaft `zoned` ist ein boolescher Wert, der beim Booten einer Zone, die ein ZFS-Dataset enthält, automatisch auf "true" gesetzt wird. Diese Eigenschaft muss vom Zonenadministrator nicht manuell gesetzt werden. Wenn die Eigenschaft `zoned` gesetzt ist, kann das Dataset in der globalen Zone nicht eingehängt bzw. für den Netzwerkzugriff freigegeben werden. Im folgenden Beispiel wurde `tank/zone/zion` an eine Zone delegiert und `tank/zone/global` nicht:

```
# 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
```

Bitte beachten Sie den Unterschied zwischen der Eigenschaft `mountpoint` und dem Verzeichnis, in das das Dataset `tank/zone/zion` gegenwärtig eingehängt ist. Die Eigenschaft `mountpoint` zeigt an, wo das Dataset auf dem Datenträger gespeichert ist, und nicht, wo es gegenwärtig eingehängt ist.

Beim Entfernen eines Datasets aus einer Zone bzw. Löschen einer Zone wird die Eigenschaft `zoned` *nicht* automatisch auf "false" zurückgesetzt. Dieses Verhalten resultiert aus den diesen Aufgaben innewohnenden Sicherheitsrisiken. Da ein nicht vertrauenswürdiger Benutzer vollständigen Zugriff auf das Dataset und seine untergeordneten Datasets hatte, kann die Eigenschaft `mountpoint` auf ungültige Werte gesetzt worden sein oder es können in den Dateisystemen `setuid`-Binärdateien vorhanden sein.

Zum Vermeiden versehentlicher Sicherheitsrisiken muss die Eigenschaft `zoned` vom Administrator der globalen Zone manuell zurückgesetzt werden, wenn das betreffende Dataset wiederverwendet werden soll. Bevor Sie die Eigenschaft `zoned` auf `off` setzen, müssen Sie sich vergewissern, dass die Eigenschaft `mountpoint` des Datasets und aller seiner untergeordneten Datasets auf zulässige Werte gesetzt ist und keine `setuid`-Binärdateien vorhanden sind, oder die Eigenschaft `setuid` deaktivieren.

Nachdem Sie sich überzeugt haben, dass keine Sicherheitslücken vorhanden sind, können Sie die Eigenschaft `zoned` mithilfe des Befehls `zfs set` oder `zfs inherit` deaktivieren. Wenn die Eigenschaft `zoned` deaktiviert wird, wenn das betreffende Dataset innerhalb einer Zone verwendet wird, kann das System unvorhersehbar reagieren. Deswegen sollten Sie den Wert dieser Eigenschaft nur dann ändern, wenn Sie sich sicher sind, dass das Dataset nicht mehr von einer nicht-globalen Zone verwendet wird.

Verwenden von ZFS-Speicher-Pools mit alternativem Root-Verzeichnis

Beim Erstellen eines Pools wird dieser inhärent mit dem Host-System verknüpft. Das Host-System verwaltet die Informationen des Pools und erkennt, wenn dieser nicht verfügbar ist. Obwohl für den Normalbetrieb nützlich, können diese Informationen hinderlich sein, wenn Sie das System von einem alternativen Datenträger booten oder einen Pool auf Wechsel-Datenträgern erstellen. Zur Lösung dieses Problems bietet ZFS *Speicher-Pools mit alternativem Root-Verzeichnis.* Speicher-Pools mit alternativem Root-Verzeichnis gelten nur temporär für einen Systemneustart, und alle Einhängepunkte werden relativ zum neuen Root-Verzeichnis des betreffenden Pools geändert.

Erstellen von ZFS-Speicher-Pools mit alternativem Root-Verzeichnis

Speicher-Pools mit alternativem Root-Verzeichnis werden am häufigsten für Wechsel-Datenträger verwendet. In einem solchen Fall wird ein einziges Dateisystem benötigt, und es soll in jede beliebige Stelle auf dem Zielsystem eingehängt werden können. Wenn mithilfe der Option `zpool create -R` ein Speicher-Pool mit alternativem Root-Verzeichnis erstellt wird, wird der Einhängepunkt des Root-Dateisystems automatisch auf `/` gesetzt, was dem Wert des alternativen Root-Verzeichnisses entspricht.

Im folgenden Beispiel wird ein Pool namens `morpheus` mit `/mnt` als alternativem Root-Verzeichnis erstellt:

```
# zpool create -R /mnt morpheus c0t0d0
# zfs list morpheus
NAME                    USED  AVAIL  REFER  MOUNTPOINT
morpheus               32.5K  33.5G     8K  /mnt
```

Bitte beachten Sie das einzige Dateisystem `morpheus`, dessen Einhängepunkt das alternative Root-Verzeichnis `/mnt` ist. Der auf Festplatte gespeicherte Einhängepunkt ist `/`, und der vollständige Pfad zu `/mnt` wird nur im ursprünglichen Kontext der Pool-Erstellung interpretiert. Dieses Dateisystem kann dann auf einem anderen System unter einem beliebigen Speicher-Pool mit alternativem Root-Verzeichnis exportiert und importiert werden, indem die Syntax `-R` *alternativer Root-Wert* verwendet wird.

```
# 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
```

Importieren von Speicher-Pools mit alternativem Root-Verzeichnis

Pools können auch mithilfe eines alternativen Root-Verzeichnisses importiert werden. Dies ist in Situationen bei der Datenwiederherstellung nützlich, wo Einhängepunkte nicht im Kontext des aktuellen Root-Verzeichnisses interpretiert werden sollen, sondern relativ zu einem temporären Verzeichnis, in dem Reparaturen ausgeführt werden können. Diese Funktion kann auch beim Einhängen von Wechseldatenträgern verwendet werden, wie im vorherigen Abschnitt beschrieben ist.

Im folgenden Beispiel wird ein Pool namens `morpheus` mit `/mnt` als alternativem Root-Verzeichnis importiert. Es wird vorausgesetzt, dass `morpheus` vorher exportiert wurde.

```
# 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
```

◆ ◆ ◆ KAPITEL 10

Problembehebung und Pool-Wiederherstellung in Oracle Solaris ZFS

Dieses Kapitel enthält Informationen zum Erkennen und Beseitigen von ZFS-Fehlern. Darüber hinaus werden hier auch Maßnahmen zum Vermeiden von Fehlfunktionen beschrieben.

Dieses Kapitel enthält die folgenden Abschnitte:

Informationen zur Wiederherstellung des vollständigen Root-Pools finden Sie in „Wiederherstellen von ZFS-Root-Pools oder Root-Pool-Snapshots“ auf Seite 169.

Identifizieren von ZFS-Problemen

Da ZFS ein Dateisystem mit DatenträgerAdministrationsfunktionen ist, können in diesem System viele verschiedene Fehler auftreten. In diesem Kapitel wird dargelegt, wie Sie allgemeine Hardwarefehler diagnostizieren und wie Sie Probleme mit Poolgeräten und Dateisystemen lösen. Folgende Typen von Problemen können auftreten:

- **Allgemeine Hardwareprobleme** – Hardwareprobleme können sich auf die Poolleistung und die Verfügbarkeit Ihrer Pooldaten auswirken. Ermitteln Sie allgemeine Hardwareprobleme wie fehlerhafte Komponenten und Speicher, bevor Sie Probleme auf einer höheren Ebene ermitteln, wie in den Pools und Dateisystemen.
- **Probleme mit ZFS-Speicherpools**
  - „Identifizieren von Problemen mit ZFS-Speicherpools“ auf Seite 298
  - „Abhilfe bei Problemen mit ZFS-Speichergeräten“ auf Seite 303

- „Abhilfe bei Problemen mit ZFS-Dateisystemen“ auf Seite 318
- „Reparieren einer beschädigten ZFS-Konfiguration“ auf Seite 328
- „Reparieren eines Systems, das nicht hochgefahren werden kann“ auf Seite 328

Bitte beachten Sie, dass in einem einzigen Pool alle drei Fehlertypen auftreten können. Deswegen umfasst eine vollständige Problembehebungsroutine das Finden und Beheben eines Fehlers, Weitergehen zum nächsten Fehler usw.

Abhilfe bei allgemeinen Hardwareproblemen

Bestimmen Sie anhand der folgenden Abschnitte, ob Poolprobleme oder die Nichtverfügbarkeit des Dateisystems auf ein Hardwareproblem zurückzuführen sind, wie Fehler bei Hauptplatine, Speicher, Geräten, HBA oder falsche Konfiguration.

Beispiel: Eine ausgefallene oder fehlerhafte Festplatte in einem belegten ZFS-Pool kann die Gesamtsystemleistung stark beeinträchtigen.

Wenn Sie als Erstes Hardwareprobleme diagnostizieren und identifizieren, die einfacher zu erkennen sind, können Sie zur Diagnose der Pool- und Dateisystemprobleme übergehen, wie im weiteren Verlauf dieses Kapitels beschrieben. Wenn die Hardware-, Pool- und Dateisystemkonfigurationen in Ordnung sind, sollten Sie Anwendungsprobleme diagnostizieren, die im Allgemeinen komplexer zu erkennen sind und in diesem Handbuch nicht abgedeckt werden.

Identifizieren von Hardware- und Gerätefehlern

Der Solaris Fault Manager überwacht Software-, Hardware- und spezifische Geräteprobleme, indem Fehlertelemetrieinformationen, die auf ein bestimmtes Symptom verweisen, im Fehlerlog identifiziert werden, und indem dann ein Bericht über die tatsächliche Fault-Diagnose erstellt wird, wenn das Fehlersymptom zu einem echten Fault führt.

Mit dem folgenden Befehl werden software- und hardwarebezogene Faults identifiziert.

```
# fmadm faulty
```

Verwenden Sie den obigen Befehl routinemäßig, um fehlerhafte Dienste oder Geräte zu identifizieren.

Verwenden Sie den obigen Befehl routinemäßig, um hardware- oder gerätebezogene Fehler zu identifizieren.

```
# fmdump -eV | more
```

Fehlermeldungen in dieser Logdatei, die `vdev.open_failed`-, `checksum`- oder `io_failure`-Probleme beschreiben, erfordern ein sofortiges Eingreifen, sonst können sie zu echten Faults werden, die mit dem Befehl `fmadm` angezeigt werden.

Wenn mit dem obigen Befehl angegeben wird, dass ein Gerät einen Fehler aufweist, muss zu diesem Zeitpunkt geprüft werden, ob ein Ersatzgerät verfügbar ist.

Mit dem Befehl `iostat` können Sie auch zusätzliche Gerätefehler überwachen. Mit der folgenden Syntax können Sie eine Zusammenfassung der Fehlerstatistiken anzeigen.

```
# iostat -en
  ---- errors ---
  s/w h/w trn tot device
    0   0   0   0 c0t5000C500335F95E3d0
    0   0   0   0 c0t5000C500335FC3E7d0
    0   0   0   0 c0t5000C500335BA8C3d0
    0  12   0  12 c2t0d0
    0   0   0   0 c0t5000C500335E106Bd0
    0   0   0   0 c0t50015179594B6F11d0
    0   0   0   0 c0t5000C500335DC60Fd0
    0   0   0   0 c0t5000C500335F907Fd0
    0   0   0   0 c0t5000C500335BD117d0
```

Die obige Ausgabe enthält einen Bericht zu Fehlern bei einer internen Festplatte `c2t0d0` . Verwenden Sie die folgende Syntax, um detailliertere Gerätefehler anzuzeigen.

```
# iostat -En
c0t5000C500335F95E3d0 Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST930003SSUN300G Revision: 0B70 Serial No: 110672QFSB
Size: 300.00GB <300000000000 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
c0t5000C500335FC3E7d0 Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST930003SSUN300G Revision: 0B70 Serial No: 110672TE67
Size: 300.00GB <300000000000 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
c0t5000C500335BA8C3d0 Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST930003SSUN300G Revision: 0B70 Serial No: 110672SDF4
Size: 300.00GB <300000000000 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
c2t0d0           Soft Errors: 0 Hard Errors: 12 Transport Errors: 0
Vendor: AMI      Product: Virtual CDROM    Revision: 1.00 Serial No:
Size: 0.00GB <0 bytes>
Media Error: 0 Device Not Ready: 12 No Device: 0 Recoverable: 0
Illegal Request: 2 Predictive Failure Analysis: 0
```

Systemprotokoll mit ZFS-Fehlermeldungen

Neben der kontinuierlichen Verfolgung von Fehlern innerhalb eines Pools zeigt ZFS beim Auftreten bestimmter Ereignisse auch Systemprotokollmeldungen an. In den folgenden Situationen werden Benachrichtigungsereignisse ausgelöst:

- **Statusübergang eines Datenspeichergeräts** – Wenn ein Datenspeichergerät in den Status `FAULTED` übergeht, protokolliert ZFS eine Meldung, die darauf hinweist, dass die Fehlertoleranz des Pools beeinträchtigt werden könnte. Eine ähnliche Meldung wird gesendet, wenn das Gerät später wieder in Betrieb genommen und die ordnungsgemäße Pool-Funktion wiederhergestellt wird.
- **Datenbeschädigung** – Beim Erkennen von Datenbeschädigungen protokolliert ZFS eine Meldung, die beschreibt, wann und wo die Datenbeschädigung erkannt wurde. Diese Meldung wird nur beim allerersten Auftreten des Ereignisses protokolliert. Ein nachfolgendes Auftreten dieser Ereignisses löst keine Meldung mehr aus.
- **Pool- und Geräteausfälle** – Wenn ein Pool oder Gerät ausfällt, meldet der FehlerAdministrationsdämon diese Fehler mithilfe von Systemprotokollmeldungen und mit dem Befehl `fmdump`.

Wenn ZFS einen Gerätefehler erkennt und diesen automatisch behebt, wird keine Benachrichtigung gesendet. Solche Fehler stellen keine Einschränkung der Pool-Redundanz bzw. Datenintegrität dar und sind darüber hinaus normalerweise die Folge eines Treiberproblems mit eigenen entsprechenden Fehlermeldungen.

Identifizieren von Problemen mit ZFS-Speicherpools

In den folgenden Abschnitten wird beschrieben, wie Sie Probleme mit ZFS-Dateisystem oder Speicher-Pools erkennen und beheben können:

- „Ermitteln, ob in einem ZFS-Speicher-Pool Probleme vorhanden sind“ auf Seite 300
- „Überprüfen der Ausgabe des Befehls `zpool status`“ auf Seite 300
- „Systemprotokoll mit ZFS-Fehlermeldungen“ auf Seite 297

Die folgenden Leistungsmerkmale dienen zur Problemerkennung in ZFS-Konfigurationen:

- Mithilfe des Befehls `zpool status` können ausführliche Informationen zum ZFS-Speicher-Pool angezeigt werden.
- Pool- und Gerätefehler werden mit ZFS/FMA-Diagnosemeldungen gemeldet.
- Frühere ZFS-Befehle, durch die Informationen zum Pool-Status geändert wurden, können mithilfe des Befehls `zpool history` angezeigt werden.

Die meisten ZFS-Probleme können mithilfe des Befehls `zpool status` erkannt werden. Mithilfe dieses Befehls werden verschiedene Fehlfunktionen im System analysiert, die wichtigsten Probleme erkannt und Empfehlungen zu Abhilfemaßnahmen sowie Verweise auf entsprechende Artikel in der Sun Knowledge Base angezeigt. Beachten Sie, dass der Befehl nur ein einziges Problem im Pool erkennen kann, obwohl mehrere Probleme vorhanden sein können. Bei Datenbeschädigungsfehlern wird beispielsweise stets vorausgesetzt, dass ein Datenspeichergerät ausgefallen ist. Durch den Austausch des ausgefallenen Geräts werden jedoch möglicherweise nicht alle Datenbeschädungsprobleme behoben.

Außerdem diagnostiziert und meldet ein ZFS-Diagnoseprogramm Pool- und Datenträgerausfälle. Darüber hinaus werden mit solchen Ausfällen im Zusammenhang stehende Prüfsummen-, E/A-, Geräte- und Poolfehler gemeldet. Von `fmd` gemeldete ZFS-Fehler werden auf der Konsole angezeigt und in der Systemprotokolldatei festgehalten. In den meisten Fällen verweist Sie die `fmd`-Meldung auf den Befehl `zpool status`, mit dessen Hilfe Sie das Problem weiter verfolgen können.

Der grundlegende Problembehebungsvorgang läuft wie folgt ab:

- Suchen Sie, falls möglich, mit dem Befehl `zpool history` die früheren ZFS-Befehle, die vor Auftreten des Problems ausgeführt wurden. Beispiel:

  ```
  # zpool history tank
  History for 'tank':
  2012-11-12.13:01:31 zpool create tank mirror c0t1d0 c0t2d0 c0t3d0
  2012-11-12.13:28:10 zfs create tank/eric
  2012-11-12.13:37:48 zfs set checksum=off tank/eric
  ```

  Beachten Sie, dass in dieser Befehlsausgabe die Prüfsummen für das Dateisystem `tank/eric` deaktiviert sind. Diese Konfiguration wird nicht empfohlen.
- Suchen Sie die Fehler in den `fmd`-Meldungen, die an der Systemkonsole bzw. in der Datei unter `/var/adm/messages` angezeigt werden.
- Weitere Reparaturanweisungen finden Sie mithilfe des Befehls `zpool status -x`.
- Beheben Sie die Probleme, indem Sie wie folgt vorgehen:
  - Ersetzen Sie das nicht verfügbare oder fehlende Gerät durch ein neues Gerät, und setzen Sie das neue Gerät in Betrieb.
  - Stellen Sie mithilfe einer Sicherungskopie die fehlerhafte Konfiguration bzw. die beschädigten Daten wieder her.
  - Überprüfen Sie die Wiederherstellung mithilfe des Befehls `zpool status -x`.
  - Erstellen Sie eine Sicherungskopie der wiederhergestellten Konfiguration (falls möglich).

In diesem Abschnitt wird beschrieben, wie Sie die Ausgabe des Befehls `zpool status` interpretieren, damit Sie Fehler diagnostizieren können. Obwohl die meisten Aufgaben automatisch mithilfe des Befehls ausgeführt werden, müssen Sie genau wissen, um welche Probleme es sich handelt, damit Sie den Ausfall diagnostizieren können. In den nachfolgenden Abschnitten wird beschrieben, wie Sie verschiedenen vorgefundene Probleme beheben können.

Ermitteln, ob in einem ZFS-Speicher-Pool Probleme vorhanden sind

Mit dem Befehl `zpool status -x` können Sie am einfachsten herausfinden, ob in einem System Probleme vorliegen. Mithilfe dieses Befehls werden nur Pools angezeigt, die problembehaftet sind. Wenn in einem System alle Pools ordnungsgemäß funktionieren, wird nur Folgendes angezeigt:

```
# zpool status -x
all pools are healthy
```

Ohne das Flag `-x` zeigt der Befehl die gesamten Statusinformationen aller Pools (oder eines in der Befehlszeile angegebenen Pools) an, auch wenn diese ordnungsgemäß funktionieren.

Weitere Informationen zu Befehlszeilenoptionen des Befehls `zpool status` finden Sie unter „Abfragen des Status von ZFS-Speicher-Pools“ auf Seite 88.

Überprüfen der Ausgabe des Befehls `zpool status`

Die gesamte Ausgabe des Befehls `zpool status` sieht ungefähr wie folgt aus:

```
# 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
 scan: scrub repaired 0 in 0h3m with 0 errors on Mon Nov 12 15:17:02 2012
config:

        NAME        STATE     READ WRITE CKSUM
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c1t1d0  ONLINE       0     0     0
            c1t2d0  UNAVAIL      0     0     0  cannot open

errors: No known data errors
```

Im folgenden Abschnitt wird diese Ausgabe beschrieben.

Gesamtinformationen zum Pool-Status

Dieser Abschnitt der Ausgabe des Befehls `zpool status` enthält die folgenden Felder (einige dieser Felder werden nur angezeigt, wenn im Pool Probleme auftreten):

`pool` Gibt den Namen des Pools an.

`state` Zeigt den aktuellen Funktionsstatus des Pools an. Diese Informationen beziehen sich lediglich auf die Fähigkeit des Pools, für die erforderliche Replikation zu sorgen.

`status` Beschreibt, was mit dem Pool nicht in Ordnung ist. Dieses Feld wird nicht angezeigt, wenn keine Fehler gefunden wurden.

`action` Eine empfohlene Aktion zur Fehlerbehebung. Dieses Feld wird nicht angezeigt, wenn keine Fehler gefunden wurden.

`see` Verweist auf einen Artikel in der Sun Knowledge Base, der ausführliche Reparaturinformationen enthält. Online-Artikel werden öfter als dieses Handbuch aktualisiert und enthalten stets die aktuellsten Reparaturanweisungen. Dieses Feld wird nicht angezeigt, wenn keine Fehler gefunden wurden.

`scrub` Zeigt den aktuellen Status einer Bereinigung an (Datum und Uhrzeit der letzten Bereinigung, Informationen zu einer laufenden Bereinigung, Informationen zur Anforderung von Bereinigungen).

`errors` Zeigt bekannte bzw. unbekannte Datenfehler an.

Informationen zur Konfiguration des ZFS-Speicherpools

Das Feld `config` in der Ausgabe des Befehls `zpool status` beschreibt die Konfigurationsstruktur der Datenspeichergeräte, die den Pool bilden, sowie deren Status und die von diesen Geräten herrührenden Fehler. Folgende Status sind möglich: `ONLINE`, `FAULTED`, `DEGRADED` oder `SUSPENDED`. Wenn der Status eines Pools nicht `ONLINE` ist, wurde die Fehlertoleranz des Pools eingeschränkt.

Im zweiten Abschnitt der Konfigurationsinformationen wird die Fehlerstatistik angezeigt. Diese Fehler werden in drei Kategorien eingeteilt:

- `READ` – E/A-Fehler während der Ausgabe einer Leseanforderung
- `WRITE` – E/A-Fehler während der Ausgabe einer Schreibanforderung
- `CKSUM` – Prüfsummenfehler, d. h., das Gerät hat nach einer Leseanforderung beschädigte Daten zurückgegeben.

Mithilfe dieser Kategorien kann ermittelt werden, ob es sich um bleibende Schäden handelt. Eine geringe Anzahl auftretender E/A-Fehler kann die Folge zeitweiliger Ausfälle sein, während eine größere Anzahl an E/A-Fehlern auf ein bleibendes Problem mit dem entsprechenden Gerät hinweisen kann. Bei diesen Fehlern muss es sich nicht unbedingt um Datenbeschädigung handeln, auch wenn sie von den Anwendungen so interpretiert werden. Wenn das Gerät zu einer redundanten Konfiguration gehört, kann es sein, dass die Datenträger nicht behebbare Fehler aufweisen, während auf der RAID-Z- bzw. Datenspiegelungsebene keine Fehler angezeigt werden. In solchen Fällen hat ZFS die unbeschädigten Daten erfolgreich abgerufen und versucht, die beschädigten Daten durch vorhandene Replikationen zu ersetzen.

Weitere Informationen zur Interpretation dieser Fehler finden Sie unter „Ermitteln des Gerätefehlertyps“ auf Seite 307.

Zusätzliche hilfreiche Informationen werden in der letzten Spalte der Ausgabe des Befehls `zpool status` angezeigt. Diese Information ergänzen die im Feld `state` enthaltenen Informationen und helfen bei der Diagnose von Fehlern. Bei Datenspeichergeräten mit dem Status `UNAVAIL` zeigt dieses Feld an, ob auf das betreffende Gerät zugegriffen werden kann oder die Daten auf dem Gerät beschädigt sind. Wenn auf das Datenspeichergerät mithilfe von Resilvering Daten neu aufgespielt werden, zeigt dieses Feld den Verlauf dieses Vorgangs an.

Weitere Informationen zur Überwachung des Resilvering-Vorgangs finden Sie in „Anzeigen des Resilvering-Status“ auf Seite 316.

Bereinigungsstatus des ZFS-Speicherpools

Im Bereinigungsabschnitt der Ausgabe des Befehls `zpool status` wird der aktuelle Status von Bereinigungsvorgängen angezeigt, die explizit ausgeführt werden. Diese Informationen weisen nicht auf Fehler hin, die im System aufgetreten sind, können aber zum Ermitteln der Genauigkeit des Meldens von Datenbeschädigungsfehlern herangezogen werden. Wenn die letzte Bereinigung erst vor kurzem ausgeführt wurde, sind Datenbeschädigungen höchstwahrscheinlich bereits bekannt.

Meldungen zum Abschluss von Bereinigungen bleiben nach Systemneustarts erhalten.

Weitere Informationen zur Datenbereinigung und zur Interpretation dieser Informationen finden Sie unter „Überprüfen der Integrität des ZFS-Dateisystems“ auf Seite 318.

ZFS-Datenbeschädigungsfehler

Der Befehl `zpool status` zeigt auch an, ob im Zusammenhang mit einem Pool bekannte Fehler aufgetreten sind. Diese Fehler können während der Datenbereinigung oder im Normalbetrieb gefunden worden sein. ZFS führt ein kontinuierliches Protokoll aller mit einem Pool im Zusammenhang stehenden Datenfehler. Nach jedem Abschluss einer vollständigen Systembereinigung wird das Protokoll entsprechend aktualisiert.

Datenbeschädigungsfehler sind stets schwerwiegend. Ihr Vorhandensein weist darauf hin, dass bei mindestens einem Anwendungsprogramm aufgrund beschädigter Daten im Pool ein E/A-Fehler aufgetreten ist. Gerätefehler innerhalb eines redundanten Pools verursachen keine Datenbeschädigung und werden in diesem Protokoll nicht festgehalten. Standardmäßig wird nur die Anzahl der gefundenen Fehler angezeigt. Eine vollständige Liste mit Fehlern und deren Informationen kann mit dem Befehl `zpool status -v` angezeigt werden. Beispiel:

```
# 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
```

Eine ähnliche Meldung wird auch von `fmd` auf der Systemkonsole angezeigt und in der Datei `/var/adm/messages` protokolliert. Diese Meldungen können auch mit dem Befehl `fmdump` verfolgt werden.

Weitere Informationen zur Interpretation von Datenbeschädigungsfehlern finden Sie unter „Ermitteln der Art der Datenbeschädigung“ auf Seite 324.

Abhilfe bei Problemen mit ZFS-Speichergeräten

In den folgenden Abschnitten wird beschrieben, wie Sie Probleme mit einem fehlenden, entfernten oder fehlerhaften Gerät beheben.

Abhilfe bei fehlendem oder entferntem Gerät

Wenn auf ein Gerät nicht zugegriffen werden kann, wird es in der Ausgabe des Befehls `zpool status` mit dem Status `UNAVAIL` angezeigt. Dieser Status bedeutet, dass ZFS beim ersten Zugriff auf den Pool nicht auf das betreffende Datenspeichergerät zugreifen konnte oder es seitdem nicht mehr verfügbar ist. Wenn durch dieses Datenspeichergerät ein virtuelles Gerät der obersten Hierarchieebene nicht mehr verfügbar ist, können von diesem Pool keine Daten abgerufen werden. Andernfalls kann die Fehlertoleranz des Pools beeinträchtigt werden. In beiden Fällen muss das Gerät dem System nur wieder zugeordnet werden, um den normalen Betrieb wiederherzustellen. Wenn Sie ein Gerät ersetzen müssen, das sich wegen eines Fehlers im Status `UNAVAIL` befindet, wird auf „Austauschen eines Datenspeichergeräts in einem ZFS-Speicher-Pool“ auf Seite 309 verwiesen.

Wenn sich ein Gerät in einem Root-Pool oder einem gespiegelten Root-Pool im Status `UNAVAIL` wird auf folgende Referenzen verwiesen:

- **Fehlerhafte Festplatte bei einem gespiegelten Root-Pool** – „Booten von einer alternativen Festplatte in einem gespiegelten ZFS-Root-Pool“ auf Seite 162

- **Ersetzen einer Festplatte in einem Root-Pool**
  - „So ersetzen Sie eine Festplatte im ZFS-Root-Pool“ auf Seite 170
- **Vollständige Notfallwiederherstellung eines Root-Pools** – „Wiederherstellen von ZFS-Root-Pools oder Root-Pool-Snapshots“ auf Seite 169

Nach einem Geräteausfall wird von `fmd` in etwa die folgende Meldung angezeigt:

```
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: daleks
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.
```

Um ausführlichere Informationen zu Geräteproblemen und deren Behebung anzuzeigen, verwenden Sie den Befehl `zpool status -x`. Beispiel:

```
# 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
  scan: scrub repaired 0 in 0h0m with 0 errors on Tue Sep 27 16:59:07 2011
config:

        NAME        STATE     READ WRITE CKSUM
        tank        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c2t2d0  ONLINE       0     0     0
            c2t1d0  UNAVAIL      0     0     0  cannot open

errors: No known data errors
```

Diese Befehlsausgabe zeigt, dass das Gerät `c2t1d0` nicht funktioniert. Wenn das Gerät fehlerhaft ist, ersetzen Sie es.

Falls erforderlich setzen Sie das ersetzte Gerät dann mit dem Befehl `zpool online` online. Beispiel:

```
# zpool online tank c2t1d0
```

Teilen Sie FMA mit, dass das Gerät ersetzt wurde, wenn die Ausgabe von `fmadm faulty` den Gerätefehler identifiziert. Beispiel:

```
# fmadm faulty
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID         SEVERITY
--------------- ------------------------------------  -------------- ---------
Sep 27 16:58:50 e6bb52c3-5fe0-41a1-9ccc-c2f8a6b56100  ZFS-8000-D3    Major

Host        : neo
Platform    : SUNW,Sun-Fire-T200        Chassis_id  :
Product_sn  :

Fault class : fault.fs.zfs.device
Affects     : zfs://pool=tank/vdev=c75a8336cda03110
                  faulted and taken out of service
Problem in  : zfs://pool=tank/vdev=c75a8336cda03110
                  faulted and taken out of service

Description : A ZFS device failed.  Refer to http://sun.com/msg/ZFS-8000-D3 for
              more information.

Response    : No automated response will occur.

Impact      : Fault tolerance of the pool may be compromised.

Action      : Run 'zpool status -x' and replace the bad device.

# fmadm repaired zfs://pool=tank/vdev=c75a8336cda03110
```

Als Letztes stellen Sie sicher, dass der Pool mit dem ersetzten Datenspeichergerät ordnungsgemäß funktioniert. Beispiel:

```
# zpool status -x tank
pool 'tank' is healthy
```

Abhilfe bei entferntem Gerät

Wenn ein Datenspeichergerät vollständig aus dem System entfernt wird, erkennt ZFS, dass es nicht geöffnet werden kann und versetzt es in den Zustand `REMOVED`. Je nach der Datenreplikationsebene des betreffenden Pools kann es sein, dass durch das Entfernen der gesamte Pool nicht mehr zur Verfügung steht. Bei der Entfernung eines Datenträgers in Konfigurationen mit Datenspiegelung oder RAID-Z bleibt der Pool weiterhin verfügbar. Ein Pool kann in den Zustand `REMOVED` versetzt werden, was bedeutet, dass Daten erst dann wieder verfügbar sind, wenn das Gerät wieder eingebunden wurde. Dies gilt unter folgenden Bedingungen:

- wenn alle Komponenten einer Spiegelung entfernt werden
- wenn mehrere Geräte in einem RAID-Z-Gerät (`raidz1`) entfernt werden
- wenn ein Gerät der obersten Hierarchieebene in einer Konfiguration mit einer Festplatte entfernt wird

Wiedereinbinden eines Datenspeichergeräts

Die Art und Weise, wie ein fehlendes Datenspeichergerät wieder in ein System eingebunden wird, hängt vom jeweiligen Gerät ab. Wenn auf das Gerät über das Netzwerk zugegriffen wird, muss die Netzwerkverbindung wiederhergestellt werden. Wenn das Datenspeichergerät ein USB-Gerät oder ein anderer Wechseldatenträger ist, muss es wieder an das System angeschlossen werden. Wenn es sich bei dem betreffenden Gerät um eine lokale Festplatte handelt, kann es sein, dass das Gerät aufgrund eines Controller-Ausfalls nicht mehr vom System erkannt wird. In diesem Fall muss der Controller ausgetauscht werden, wonach die betreffenden Datenträger wieder verfügbar sind. Es können noch weitere Probleme vorliegen, die mit der der Art der Hardware und ihrer Konfiguration in Zusammenhang stehen. Wenn ein Laufwerk ausfällt und vom System nicht mehr erkannt wird, muss das Datenspeichergerät als beschädigt eingestuft werden. Folgen Sie der unter „Ersetzen oder Reparieren eines beschädigten Geräts“ auf Seite 307 beschriebenen Vorgehensweise.

ein Pool könnte sich im Status `SUSPENDED` befinden, wenn die Verbindung zu dem Gerät unterbrochen ist. Ein `SUSPENDED` Pool bleibt im Wartezustand, bis das Geräteproblem behoben ist. Beispiel:

```
# zpool status cybermen
  pool: cybermen
 state: SUSPENDED
status: One or more devices are unavailable in response to IO failures.
        The pool is suspended.
action: Make sure the affected devices are connected, then run 'zpool clear' or
        'fmadm repaired'.
    see: http://www.sun.com/msg/ZFS-8000-HC
  scan: none requested
config:

        NAME           STATE     READ WRITE CKSUM
        cybermen       UNAVAIL      0    16     0
            c8t3d0     UNAVAIL      0     0     0
            c8t1d0     UNAVAIL      0     0     0
```

Nachdem die Verbindung zu dem Gerät wiederhergestellt ist, setzen Sie die Pool- oder Gerätefehler zurück.

```
# zpool clear cybermen
# fmadm repaired zfs://pool=name/vdev=guid
```

Benachrichtigung von ZFS nach Wiederherstellung der Verfügbarkeit

Nach dem Wiedereinbinden eines Datenspeichergeräts in das System erkennt ZFS unter Umständen seine Verfügbarkeit automatisch. Dies muss aber nicht so ein. Wenn der Pool vorher `UNAVAIL` oder `SUSPENDED` war oder das System während der `attach`-Prozedur neu gestartet wurde, sucht ZFS automatisch alle Datenspeichergeräte ab, während es versucht, auf den Pool zuzugreifen. Wenn der Pool in seiner Funktionstüchtigkeit beeinträchtigt war und das

Gerät während des Systembetriebs ausgetauscht wurde, müssen Sie ZFS mithilfe des Befehls `zpool online` benachrichtigen, dass das Gerät jetzt wieder verfügbar ist und wieder auf das Gerät zugegriffen werden kann. Beispiel:

```
# zpool online tank c0t1d0
```

Weitere Informationen zum Inbetriebnehmen von Geräten finden Sie unter „Inbetriebnehmen eines Gerätes“ auf Seite 76.

Ersetzen oder Reparieren eines beschädigten Geräts

In diesem Abschnitt wird beschrieben, wie die verschiedenen Fehlertypen eines Datenspeichergerätes ermittelt, vorübergehende Fehler gelöscht und Geräte ausgetauscht werden können.

Ermitteln des Gerätefehlertyps

Der Begriff *beschädigtes Gerät* ist nicht klar umrissen und kann verschiedene mögliche Situationen beschreiben:

- **Bitfäule** – Mit der Zeit können äußere Einflüsse wie Magnetfelder und kosmische Strahlung dazu führen, dass auf Datenträgern gespeicherte Bits unvorhergesehene Werte annehmen. Solche Ereignisse sind relativ selten, treten aber häufig genug auf, um potenzielle Datenbeschädigung in größeren und lange laufenden Systemen zu verursachen.
- **Fehlgeleitete Lese- oder Schreibvorgänge** – Firmware- oder Hardwarebug können dazu führen, dass Lese- oder Schreibvorgänge ganzer Datenblöcke auf den falschen Bereich auf dem Datenträger verweisen. Diese Fehler sind normalerweise vorübergehend, obwohl eine große Anzahl solcher Fehler auf ein fehlerhaftes Laufwerk hinweisen kann.
- **Administratorfehler** – Administratoren können versehentlich Datenträgerbereiche mit ungültigen Daten überschreiben (z. B. das Kopieren von `/dev/zero` auf bestimmte Datenträgerbereiche) und so Daten auf dem Datenträger dauerhaft beschädigen. Solche Fehler sind stets vorübergehend.
- **Zeitweilige Ausfälle** – Datenträger können zeitweilig ausfallen, wodurch E/A-Vorgänge fehlschlagen. Diese Situation tritt normalerweise bei Datenspeichergeräten auf, auf die über das Netzwerk zugegriffen wird, obwohl auch bei lokalen Datenträgern solche Ausfälle auftreten können. Solche Fehler sind können vorübergehend sein.
- **Fehlerhafte bzw. unzuverlässig arbeitende Hardware** – Unter diese Kategorie fallen alle durch fehlerhafte Hardware verursachten Probleme. Dies können dauerhafte E/A-Fehler, falscher Datentransport und daraus folgende regellose Datenbeschädigung sowie eine Reihe anderer Fehler sein. Solche Fehler sind normalerweise dauerhaft.

- **Außer Betrieb genommene Datenspeichergeräte** – Wenn ein Gerät außer Betrieb genommen wurde, wird angenommen, dass es der Administrator in diesen Zustand versetzt hat, weil es fehlerhaft ist. Der Administrator, der das Gerät in diesen Zustand versetzt hat, kann überprüfen, ob diese Annahme richtig ist.

Die genaue Ermittlung von Fehlerursachen kann sich schwierig gestalten. Der erste Schritt besteht darin, die Fehlerzähler in der Ausgabe des Befehls `zpool status` zu überprüfen. Beispiel:

```
# zpool status -v tank
  pool: tank
 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
  scan: scrub in progress since Tue Sep 27 17:12:40 2011
    63.9M scanned out of 528M at 10.7M/s, 0h0m to go
    0 repaired, 12.11% done
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       2     0     0
          mirror-0  ONLINE       2     0     0
            c2t2d0  ONLINE       2     0     0
            c2t1d0  ONLINE       2     0     0

errors: Permanent errors have been detected in the following files:

        /tank/words
```

Die Fehler werden in E/A- und Prüfsummenfehler unterteilt. Daraus lässt sich unter Umständen auf den Fehlertyp schließen. Im Normalbetrieb treten während einer langen Systemlaufzeit nur einige wenige Fehler auf. Wenn eine hohe Anzahl von Fehlern angezeigt wird, deutet dies wahrscheinlich auf einen bevorstehenden bzw. vollständigen Geräteausfall hin. Ein Administratorfehler kann jedoch ebenfalls zu hohen Fehleranzahlen führen. Eine weitere Informationsquelle ist das Systemprotokoll. Wenn das Protokoll eine große Anzahl an Meldungen von SCSI- bzw. Fibre Channel-Treibern enthält, weist dies möglicherweise auf schwerwiegende Hardwareprobleme hin. Wenn keine Systemmeldungen protokolliert werden, ist der Schaden wahrscheinlich vorübergehend.

Das Ziel besteht in der Beantwortung der folgenden Frage:

Ist es wahrscheinlich, dass an diesem Gerät wieder ein Fehler auftritt?

Nur einmal auftretende Fehler werden als *vorübergehend* eingestuft und ziehen keine potenziellen Ausfälle nach sich. Fehler, die dauerhaft oder ernstlich genug sind, um potenzielle Hardwareausfälle auszulösen, werden als *schwerwiegend* eingestuft. Die Ermittlung von Fehlertypen sprengt den Rahmen der gegenwärtig mit ZFS verfügbaren automatisierten

Softwarelösungen und muss manuell vom Administrator durchgeführt werden. Nach der Ermittlung des Fehlertyps können entsprechende Abhilfemaßnahmen ergriffen werden. Diese bestehen entweder im Löschen vorübergehender Fehler oder im Austauschen des betreffenden Datenspeichergeräts aufgrund schwerwiegender Fehler. Die entsprechenden Reparaturvorgänge werden in den nächsten Abschnitten erläutert.

Auch wenn Gerätefehler als vorübergehend eingestuft wurden, können sie trotzdem nicht mehr rückgängig zu machende Datenfehler im Pool verursacht haben. Diese Fehler erfordern auch dann spezielle Reparaturmaßnahmen, wenn das zugrunde liegende Datenspeichergerät ordnungsgemäß funktioniert oder anderweitig repariert wurde. Weitere Informationen zum Beseitigen von Datenfehlern finden Sie unter „Reparieren beschädigter Daten“ auf Seite 323.

Zurücksetzen vorübergehender Gerätefehler

Wenn Gerätefehler als vorübergehend eingestuft wurden und die zukünftige ordnungsgemäße Funktion des betreffenden Datenspeichergeräts nicht beeinträchtigen, können sie gelöscht werden. Damit wird angezeigt, dass kein schwerwiegender Fehler aufgetreten ist. Mit dem Befehl `zpool clear` können Sie Fehlerzähler für Datenspeichergeräte mit RAID-Z- bzw. Datenspiegelungskonfigurationen zurücksetzen. Beispiel:

```
# zpool clear tank c1t1d0
```

Mithilfe dieser Syntax werden alle Fehler gelöscht und alle Fehlerzähler zurückgesetzt, die mit dem betreffenden Datenspeichergerät in Zusammenhang stehen.

Zum Löschen aller mit den virtuellen Geräten im Pool in Verbindung stehenden Fehler und Zurücksetzen aller Fehlerzähler dient die folgende Syntax:

```
# zpool clear tank
```

Weitere Informationen zum Löschen von Pool-Fehlern finden Sie unter „Löschen von Gerätefehlern im Speicher-Pool“ auf Seite 77.

Austauschen eines Datenspeichergeräts in einem ZFS-Speicher-Pool

Wenn ein Datenspeichergerät dauerhaft beschädigt ist bzw. ein solcher Schaden bevorsteht, muss es ausgetauscht werden. Ob das betreffende Gerät ersetzt werden kann, hängt von der Konfiguration ab.

- „Ermitteln, ob ein Gerät ausgetauscht werden kann“ auf Seite 310
- „Datenspeichergeräte, die nicht ausgetauscht werden können“ auf Seite 310
- „Austauschen eines Datenspeichergeräts in einem ZFS-Speicher-Pool“ auf Seite 311
- „Anzeigen des Resilvering-Status“ auf Seite 316

Ermitteln, ob ein Gerät ausgetauscht werden kann

Wenn das zu ersetzende Gerät zu einer redundanten Konfiguration gehört, müssen ausreichende Replikationen vorhanden sein, aus denen unbeschädigte Daten wiederhergestellt werden können. Wenn beispielsweise zwei Datenträger in einer vierfachen Datenspiegelungskonfiguration `UNAVAIL` sind, können beide Datenträger ausgetauscht werden, da gültige Datenreplikationen vorhanden sind. Wenn jedoch zwei Datenträger in einer vierfachen RAID-Z-Konfiguration `raidz1` `UNAVAIL` sind, kann keiner der beiden Datenträger ausgetauscht werden, da nicht genügend gültige Datenreplikationen verfügbar sind, aus denen Daten wiederhergestellt werden können. Wenn das Gerät beschädigt, aber noch in Betrieb ist, kann es ausgetauscht werden, solange sich der Pool nicht im Status `UNAVAIL` befindet. Wenn nicht genügend Replikationen mit gültigen Daten verfügbar sind, werden jedoch auch die beschädigten Daten auf das neue Datenspeichergerät kopiert.

In der folgenden Konfiguration kann der Datenträger `c1t1d0` ausgetauscht werden, und die gesamten Daten im Pool werden von der ordnungsgemäßen Replikation `c1t0d0` kopiert.

```
    mirror            DEGRADED
    c1t0d0             ONLINE
    c1t1d0             FAULTED
```

Der Datenträger `c1t0d0` kann ebenfalls ausgetauscht werden, obwohl keine Datenselbstheilung erfolgen kann, da keine ordnungsgemäße Datenreplikation vorhanden ist.

In der folgenden Konfiguration kann keiner der Datenträger im Status `UNAVAIL` ausgetauscht werden. Die Datenträger mit dem Status `ONLINE` können ebenfalls nicht ausgetauscht werden, da der Pool selbst `UNAVAIL` ist.

```
    raidz              FAULTED
    c1t0d0             ONLINE
    c2t0d0             FAULTED
    c3t0d0             FAULTED
    c4t0d0             ONLINE
```

In der folgenden Konfiguration können alle Datenträger der obersten Hierarchieebene ausgetauscht werden, obwohl auch auf den alten Datenträgern befindliche ungültige Daten auf den neuen Datenträger kopiert werden.

```
c1t0d0         ONLINE
c1t1d0         ONLINE
```

Wenn einer der Datenträger `UNAVAIL` ist, kann nichts ausgetauscht werden, da dann der Pool selbst `UNAVAIL` ist.

Datenspeichergeräte, die nicht ausgetauscht werden können

Wenn ein Pool durch den Ausfall eines Datenspeichergeräts `UNAVAIL` wird oder das betreffende Gerät in einer nicht redundanten Konfiguration zu viele Datenfehler enthält, kann es nicht

sicher ausgetauscht werden. Ohne ausreichende Redundanz sind keine entsprechenden gültigen Daten vorhanden, die die Fehler auf dem beschädigten Gerät beseitigen könnten. In einem solchen Fall besteht nur die Möglichkeit, den Pool zu löschen, die Konfiguration neu zu erstellen und die Daten aus einer Sicherungskopie wiederherzustellen.

Weitere Informationen zum Wiederherstellen eines gesamten Pools finden Sie unter „Reparieren von Schäden am gesamten ZFS-Speicher-Pool“ auf Seite 326.

Austauschen eines Datenspeichergeräts in einem ZFS-Speicher-Pool

Wenn Sie ermittelt haben, dass ein Datenspeichergerät ausgetauscht werden kann, können Sie es mithilfe des Befehls `zpool replace` ersetzen. Um ein beschädigtes Gerät durch ein anderes Gerät zu ersetzen, verwenden Sie folgende Syntax:

```
# zpool replace tank c1t1d0 c2t0d0
```

Mithilfe dieses Befehls werden Daten vom beschädigten Gerät oder von anderen Geräten im Pool, sofern sich dieser in einer redundanten Konfiguration befindet, auf das neue Gerät migriert. Nach Abschluss des Befehls wird das beschädigte Datenspeichergerät von der Konfiguration abgetrennt und kann dann aus dem System entfernt werden. Wenn Sie das Datenspeichergerät bereits entfernt und an der gleichen Stelle durch ein neues Gerät ersetzt haben, sollten Sie das Einzelgeräteformat des Befehls verwenden. Beispiel:

```
# zpool replace tank c1t1d0
```

Mithilfe dieses Befehls wird das neue Datenspeichergerät formatiert, und anschließend werden die Daten aus der Konfiguration durch Resilvering aufgespielt.

Weitere Informationen zum Befehl `zpool replace` finden Sie unter „Austauschen von Geräten in einem Speicher-Pool“ auf Seite 77.

BEISPIEL 10–1 Austauschen einer SATA-Festplatte in einem ZFS-Speicherpool

Das folgende Beispiel zeigt, wie ein Gerät (`c1t3d0`) in einem Speicherpool mit Datenspiegelung `tank` auf einem System mit SATA-Geräten ersetzt wird. Um die Festplatte `c1t3d0` durch eine neue Festplatte an derselben Position (`c1t3d0`) ersetzen, müssen Sie die Festplatte zunächst dekonfigurieren. Wenn die zu ersetzende Festplatte keine SATA-Festplatte ist, wird auf „Austauschen von Geräten in einem Speicher-Pool“ auf Seite 77 verwiesen.

Der Vorgang wird im Wesentlichen wie folgt durchgeführt:

- Nehmen Sie die Festplatte (`c1t3d0`), die ersetzt werden soll, außer Betrieb. Eine in Gebrauch befindliche SATA-Festplatte kann nicht dekonfiguriert werden.

BEISPIEL 10–1 Austauschen einer SATA-Festplatte in einem ZFS-Speicherpool *(Fortsetzung)*

- Verwenden Sie den Befehl `cfgadm`, um die SATA-Festplatte (`c1t3d0`) zu ermitteln, und entfernen Sie sie aus der Konfiguration. Der Pool wird durch die außer Betrieb genommene Festplatte in der Datenspiegelungskonfiguration in einen eingeschränkten Zustand versetzt, bleibt aber weiterhin verfügbar.
- Ersetzen Sie die Festplatte physisch (`c1t3d0`). Vergewissern Sie sich vor dem Ausbauen des `UNAVAIL`-Laufwerks, dass die blaue LED `Ready to Remove` leuchtet.
- Konfigurieren Sie die SATA-Festplatte (`c1t3d0`) neu.
- Setzen Sie die neue Festplatte (`c1t3d0`) in Betrieb.
- Führen Sie den Befehl `zpool replace` aus, um die Festplatte (`c1t3d0`) zu ersetzen.

Hinweis – Wenn Sie zuvor die Eigenschaft `autoreplace` des Pools auf `on` gesetzt haben, wird jedes neue Gerät, das sich an der gleichen Stelle befindet, an der sich zuvor ein anderes zum Pool gehörendes Geräts befand, automatisch formatiert und ohne den Befehl `zpool replace` ersetzt. Dieses Leistungsmerkmal wird möglicherweise nicht auf jeder Art von Hardware unterstützt.

- Wenn eine ausgefallene Festplatte automatisch durch eine Hot-Spare-Festplatte ersetzt wurde, müssen Sie die Hot-Spare-Festplatte möglicherweise nach dem Ersetzen der ausgefallenen Festplatte abtrennen. Wenn `c2t4d0` beispielsweise noch immer eine aktive Hot-Spare-Festplatte ist, nachdem die ausgefallene Festplatte ersetzt wurde, trennen Sie sie ab.

  ```
  # zpool detach tank c2t4d0
  ```

- Wenn FMA das fehlerhafte Gerät protokolliert, müssen Sie den Gerätefehler zurücksetzen.

  ```
  # fmadm faulty
  # fmadm repaired zfs://pool=name/vdev=guid
  ```

Im folgenden Beispiel wird gezeigt, wie eine Festplatte in einem ZFS-Speicher-Pool ersetzt wird.

```
# 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
```

BEISPIEL 10–1 Austauschen einer SATA-Festplatte in einem ZFS-Speicherpool *(Fortsetzung)*

```
# 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
```

Beachten Sie, dass in der obigen `zpool`-Ausgabe sowohl die neue als auch die alte Festplatte unter *replacing* (wird ersetzt) angezeigt werden kann. Beispiel:

```
replacing     DEGRADED     0     0    0
  c1t3d0s0/o  FAULTED      0     0    0
  c1t3d0      ONLINE       0     0    0
```

Dieser Text bedeutet, dass der Austauschvorgang läuft und an der neuen Festplatte das Resilvering durchgeführt wird.

Um eine Festplatte (`c1t3d0`) durch eine andere Festplatte (`c4t3d0`) zu ersetzen, müssen Sie nur den Befehl `zpool replace` ausführen. Beispiel:

```
# 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
```

BEISPIEL 10–1 Austauschen einer SATA-Festplatte in einem ZFS-Speicherpool *(Fortsetzung)*

```
              c4t3d0     ONLINE       0     0     0

errors: No known data errors
```

Mitunter muss der Befehl `zpool status` mehrmals ausgeführt werden, bevor der Austauschvorgang abgeschlossen ist.

```
# 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
            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
```

BEISPIEL 10–2 Ersetzen eines fehlgeschlagenen Protokolliergeräts

ZFS identifiziert Intent-Logfehler in der `zpool status`-Befehlsausgabe. Diese Fehler werden auch von der Fault Management Architecture (FMA) gemeldet. Ein Intent-Protokoll-Fehler kann sowohl mit ZFS als auch mit FMA behoben werden.

Das folgende Beispiel zeigt die Wiederherstellung des Normalzustands nach dem Ausfall eines Protokolliergeräts (`c0t5d0`) im Speicher-Pool (`pool`). Der Vorgang wird im Wesentlichen wie folgt durchgeführt:

- Überprüfen Sie die Ausgabe des Befehls `zpool status -x` und die FMA-Diagnosemeldung, beschrieben unter:

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&alias=EVENT:ZFS-8000-K4`
- Ersetzen Sie das fehlgeschlagene Protokolliergerät physisch.
- Nehmen Sie das neue Protokolliergerät in Betrieb.
- Löschen Sie den Fehlerzustand des Pools.
- Setzen Sie den FMA-Fehler zurück.

```
# zpool status -x
  pool: pool
```

BEISPIEL 10–2 Ersetzen eines fehlgeschlagenen Protokolliergeräts *(Fortsetzung)*

```
 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
```

Wenn das System beispielsweise plötzlich herunterfährt, bevor synchrone Schreibvorgänge in einem Pool mit separatem Protokolliergerät festgeschrieben werden können, werden Meldungen wie die folgenden angezeigt:

```
# 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-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
# fmadm faulty
# fmadm repair zfs://pool=name/vdev=guid
```

Nach einem Ausfall des Protokolliergeräts können Sie den Normalbetrieb folgendermaßen wiederherstellen:

- Ersetzen Sie das Protokolliergerät oder stellen Sie es wieder her. In diesem Beispiel handelt es sich um das Protokolliergerät `c0t5d0`.
- Nehmen Sie das Protokolliergerät wieder in Betrieb.

BEISPIEL 10–2 Ersetzen eines fehlgeschlagenen Protokolliergeräts *(Fortsetzung)*

```
# zpool online pool c0t5d0
```

- Setzen Sie den Fehlerzustand "Protokolliergerät fehlgeschlagen" zurück.

```
# zpool clear pool
```

Wenn Sie den Normalbetrieb wiederherstellen möchten, ohne das Protokolliergerät zu ersetzen, können Sie den Fehler mit dem Befehl `zpool clear` löschen. In diesem Szenario läuft der Pool in eingeschränktem Modus, und die Protokolleinträge werden in den Haupt-Pool geschrieben, bis das separate Protokolliergerät ersetzt wurde.

Verwenden Sie gespiegelte Protokolliergeräte, um den Ausfall von Protokolliergeräten zu vermeiden.

Anzeigen des Resilvering-Status

Je nach Kapazität des Datenträgers und der Datenmenge im Pool kann das Austauschen eines Datenspeichergeräts geraume Zeit dauern. Der Vorgang des Übertragens von Daten von einem Datenspeichergerät auf ein anderes Gerät nennt man *Resilvering*. Er kann mithilfe des Befehls `zpool status` überwacht werden.

Herkömmliche Dateisysteme kopieren Daten auf Datenblockebene. Da in ZFS die künstliche Schicht des Volume Managers beseitigt wurde, kann das Resilvering hier viel leistungsfähiger und kontrollierter durchgeführt werden. Die beiden Hauptvorteile dieser Funktion bestehen in Folgendem:

- ZFS kopiert beim Resilvering nur die Mindestmenge erforderlicher Daten. Im Falle eines kurzen Ausfalls (im Gegensatz zu einem vollständigen Austausch von Datenspeichergeräten) können Daten innerhalb weniger Minuten bzw. Sekunden auf den gesamten Datenträger aufgespielt werden. Nach dem Austauschen eines Datenträgers ist der Zeitraum, den das Wiederaufspielen von Daten benötigt, proportional zur Datenmenge auf dem betreffenden Datenträger. Der Austausch eines 500-GB-Datenträgers kann in Sekundenschnelle durchgeführt werden, wenn im Pool nur einige wenige GB mit Daten belegt sind.
- Das Resilvering kann unterbrochen werden und ist sicher. Wenn am System ein Stromausfall auftritt oder es neu gestartet wird, fährt der Resilvering-Vorgang an genau der Stelle fort, wo er unterbrochen wurde, ohne dass dazu manuelle Eingriffe erforderlich sind.

Der Status des Resilvering-Vorgangs kann mit dem Befehl `zpool status` angezeigt werden. Beispiel:

```
# 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
```

In diesem Beispiel wird der Datenträger `c1t0d0` durch das Gerät `c2t0d0` ersetzt. Dieses Ereignis wird in der Statusausgabe durch das virtuelle Ersatzgerät in der Konfiguration dargestellt. Dieses Gerät ist kein echtes Gerät, und Sie können mit diesem Gerät auch keinen Pool erstellen. Der Zweck dieses Geräts besteht lediglich darin, den Resilvering-Vorgang anzuzeigen, damit festgestellt werden kann, welcher Datenträger ausgetauscht wird.

Beachten Sie, dass ein Pool, in dem ein Resilvering stattfindet, in den Status `ONLINE` oder `DEGRADED` versetzt wird, da er während des Resilvering-Vorgangs nicht das erforderliche Niveau an Datenredundanz gewährleisten kann. Das Resilvering findet so schnell wie möglich statt, obwohl die damit verbundenen E/A-Prozesse stets eine niedrigere Priorität als E/A-Benutzeranforderungen haben, um die Auswirkung auf die Systemleistung zu minimieren. Nach dem Abschluss des Resilvering wird die neue, vollständige Konfiguration vom System übernommen. Beispiel:

```
# 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
```

Der Pool befindet sich jetzt wieder im Status `ONLINE`, und der ursprüngliche ausgefallene Datenträger (`c1t0d0`) wurde aus der Konfiguration entfernt.

Abhilfe bei Problemen mit ZFS-Dateisystemen

Abhilfe bei Datenproblemen in einem ZFS-Speicherpool

Beispiele für Datenprobleme umfassen:

- vorübergehende E/A-Fehler aufgrund eines fehlerhaften Datenträgers bzw. Controllers
- Datenbeschädigung auf Datenträgern aufgrund kosmischer Strahlung
- Treiberbug, wegen denen Daten falsch transportiert werden
- versehentliches Überschreiben von Bereichen auf einem physischen Datenträger durch einen Benutzer

In einigen Fällen treten solche Fehler nur vorübergehend auf, so z. B. bei sporadischen E/A-Fehlern aufgrund eines Controller-Problems. In anderen Fällen ist der Schaden bleibend, wie z. B. bei der Datenbeschädigung auf einem Datenträger. Auch wenn der Schaden bleibend ist, heißt das nicht, dass der Fehler wiederholt auftritt. Beispiel: Wenn Sie versehentlich Bereiche einer Festplatte überschreiben, ist kein Hardwareausfall aufgetreten, und das Datenspeichergerät muss nicht ausgetauscht werden. Genau festzustellen, mit welchem Problem das Datenspeichergerät behaftet ist, stellt keine leichte Aufgabe dar und wird später eingehender behandelt.

Überprüfen der Integrität des ZFS-Dateisystems

Für ZFS gibt es kein Serviceprogramm wie `fsck`. Dieses Serviceprogramm diente üblicherweise zur Reparatur und Validierung von Dateisystemen.

Reparatur von Dateisystemen

Bei herkömmlichen Dateisystemen ist die Art und Weise des Schreibens von Daten von Natur aus anfällig für unerwartete Ausfälle, die zu Inkonsistenzen im Dateisystem führen. Da herkömmliche Dateisysteme nicht transaktionsorientiert sind, können unreferenzierte Datenblöcke, ungültige Verknüpfungszähler oder andere inkonsistente Dateisystemstrukturen auftreten. Mit der Einführung des so genannten Journaling wurden zwar einige dieser Probleme behoben, es können jedoch neue Probleme auftreten, wenn Transaktionen nicht rückgängig gemacht werden können. Inkonsistente Daten in einer ZFS-Konfiguration können nur bei Hardware-Ausfällen (was durch redundante Pools vermieden werden kann) oder bei Bugs in der ZFS-Software auftreten.

Das Serviceprogramm `fsck` beseitigt bekannte Probleme, von denen UFS-Dateisysteme betroffen sind. Die meisten Probleme, von denen ZFS-Speicher-Pools betroffen sind, sind auf ausgefallene Hardware oder Stromausfälle zurückzuführen. Viele Probleme können durch redundante Pools vermieden werden. Wenn Ihr Pool durch ausgefallene Hardware oder einen Stromausfall beschädigt wurde, gehen Sie wie unter „Reparieren von Schäden am gesamten ZFS-Speicher-Pool“ auf Seite 326 beschrieben vor.

Wenn ein Pool nicht redundant ist, besteht immer das Risiko, dass Sie nach einer Beschädigung des Dateisystems nicht mehr auf Ihre Daten zugreifen können.

Validierung von Dateisystemen

Neben der Dateisystemreparatur stellt das Serviceprogramm `fsck` sicher, dass Daten auf einem Datenträger fehlerfrei sind. Diese Validierung wird üblicherweise durch Aushängen des betreffenden Dateisystems und Ausführen des Serviceprogramms `fsck` durchgeführt. Während dieses Vorgangs muss das System möglicherweise in den Einzelbenutzermodus gebracht werden. Daraus resultieren Ausfallzeiten, die proportional zur Größe des zu überprüfenden Dateisystems sind. Statt eines Serviceprogramms zum expliziten Ausführen der entsprechenden Überprüfungen besitzt ZFS einen Mechanismus zur regelmäßigen Überprüfung auf Inkonsistenzen. Diese als *Bereinigung* bezeichnete Funktion wird häufig in Arbeitsspeichern und anderen Systemen als Methode des Erkennens und Vermeidens von Problemen eingesetzt, die Hardwareausfälle oder Softwarefehlfunktionen verursachen.

Kontrollieren der ZFS-Datenbereinigung

Wenn ZFS einen Fehler erkennt (der auf die Bereinigung oder den Zugriff auf Dateien zurückzuführen ist), wird dieser intern protokolliert, sodass Sie einen schnellen Überblick über alle bekannten Fehler im Pool erhalten.

Explizite ZFS-Datenbereinigung

Die einfachste Methode zum Überprüfen der Datenintegrität besteht im Durchführen einer expliziten Bereinigung aller im Pool enthaltenen Daten. Diesem Vorgang werden alle Daten im Pool einmalig unterzogen, wodurch sichergestellt wird, dass alle Datenblöcke gelesen werden können. Die Bereinigung erfolgt so schnell, wie es das entsprechende Datenspeichergerät zulässt, obwohl E/A-Vorgänge eine niedrigere Priorität als normale Prozessen haben. Dieser Vorgang kann sich negativ auf die Systemleistung auswirken, obgleich die Daten des Pools während der Bereinigung weiterhin verwendbar und weitgehend verfügbar bleiben sollten. Explizite Bereinigungen können mit dem Befehl `zpool scrub` ausgeführt werden. Beispiel:

```
# zpool scrub tank
```

Der Status der aktuellen Bereinigung kann mithilfe des Befehls `zpool status` angezeigt werden. Beispiel:

```
# 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
```

Pro Pool kann immer nur eine aktive Bereinigung ausgeführt werden.

Mithilfe der Option `-s` können Sie eine laufende Bereinigung stoppen. Beispiel:

```
# zpool scrub -s tank
```

In den meisten Fällen sollte eine Bereinigung vollständig abgeschlossen werden, um die Datenintegrität sicherzustellen. Falls sich eine Bereinigung negativ auf die Systemleistung auswirkt, können Sie sie jederzeit abbrechen.

Durch regelmäßige Bereinigungen wird kontinuierlicher E/A-Datenverkehr zu allen Datenträgern des Systems gewährleistet. Ein Nebeneffekt der Bereinigung besteht darin, dass die StromAdministration im Leerlauf befindliche Datenträger nicht in den Energiesparmodus schalten kann. Wenn das System E/A-Vorgänge jederzeit und ohne größere Störungen ausführen kann oder Stromverbrauch kein Problem ist, kann dieser Effekt problemlos ignoriert werden.

Weitere Informationen zur Interpretation der Ausgabe des Befehls `zpool status` finden Sie unter „Abfragen des Status von ZFS-Speicher-Pools“ auf Seite 88.

ZFS-Datenbereinigung und Resilvering

Nach dem Ersetzen eines Datenspeichergeräts wird ein so genanntes "Resilvering" (Wiederaufspielen von Daten) durchgeführt, um Daten von unbeschädigten Kopien auf den neuen Datenträger zu kopieren. Dieser Vorgang ist eine Form der Datenträgerbereinigung. Aus diesem Grunde kann in einem Pool immer nur ein solcher Vorgang stattfinden. Wenn ein Bereinigungsvorgang ausgeführt wird, unterbricht der Wiederherstellungsvorgang die aktuelle Bereinigung und startet sie nach Abschluss der Wiederherstellung neu.

Weitere Informationen zum Resilvering finden Sie unter „Anzeigen des Resilvering-Status“ auf Seite 316.

Beschädigte ZFS-Daten

Datenbeschädigung tritt auf, wenn sich Gerätefehler (die auf eines oder mehrere fehlende bzw. beschädigte Datenspeichergeräte hinweisen) auf virtuelle Geräte der obersten Hierarchieebene auswirken. So können beispielsweise in einer Hälfte einer Datenspiegelungskonfiguration Tausende Gerätefehler auftreten, ohne dass dies zur Datenbeschädigung führt. Wenn auf der anderen Seite der Spiegelung an genau derselben Stelle ein Fehler auftritt, werden die Daten beschädigt.

Eine Datenbeschädigung ist stets bleibend und muss bei der Reparatur besonders berücksichtigt werden. Auch wenn die betreffenden Datenspeichergeräte repariert oder ausgetauscht werden, sind die ursprünglichen Daten für immer verloren. Meist müssen Daten in solchen Situationen aus Sicherungskopien wiederhergestellt werden. Datenfehler werden beim Auftreten protokolliert und können durch regelmäßige Pool-Bereinigung (siehe folgender Abschnitt) in Grenzen gehalten werden. Beim Entfernen eines beschädigten Datenblocks erkennt der nächste Durchlauf der Datenträgerbereinigung, dass die Datenbeschädigung nicht mehr auf dem System vorhanden ist und entfernt die Protokollierung des Fehlers vom System.

Abhilfe bei ZFS-Speicherplatzproblemen

Prüfen Sie die folgenden Abschnitte, wenn Sie nicht sicher sind, wie ZFS die Speicherplatzberechnung bei Dateisystem und Pool protokolliert. Siehe auch „Berechnung von ZFS-Festplattenkapazität“ auf Seite 32.

Speicherplatzprotokollierung bei ZFS-Dateisystem

Mit den Befehlen `zpool list` und `zfs list` kann der verfügbare Speicherplatz im Pool und Dateisystem besser bestimmt werden als mit den früheren Befehlen `df` und `du`. Mit den Legacy-Befehlen kann nur schwierig zwischen Pool- und Dateisystemspeicherplatz unterschieden werden. Außerdem berücksichtigen die Legacy-Befehle den Speicherplatz nicht, der von untergeordneten Dateisystemen oder Schnappschüssen belegt wird.

Beispiel: Bei dem folgenden Root-Pool (`rpool`) sind 5,46 GB zugewiesen und 68,5 GB frei.

```
# zpool list rpool
NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool   74G  5.46G  68.5G   7%  1.00x  ONLINE  -
```

Wenn Sie die Speicherplatzberechnung für den Pool mit der Speicherplatzberechnung für das Dateisystem vergleichen, indem Sie die Spalte `USED` der einzelnen Dateisysteme prüfen, sehen Sie, dass der Poolspeicherplatz, der in `ALLOC` protokolliert wird, in der `USED`-Gesamtsumme des Dateisystems berücksichtigt wird. Beispiel:

```
# zfs list -r rpool
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    5.41G  67.4G  74.5K  /rpool
rpool/ROOT               3.37G  67.4G    31K  legacy
rpool/ROOT/solaris       3.37G  67.4G  3.07G  /
rpool/ROOT/solaris/var    302M  67.4G   214M  /var
rpool/dump               1.01G  67.5G  1000M  -
rpool/export             97.5K  67.4G    32K  /rpool/export
rpool/export/home        65.5K  67.4G    32K  /rpool/export/home
rpool/export/home/admin  33.5K  67.4G  33.5K  /rpool/export/home/admin
rpool/swap               1.03G  67.5G  1.00G  -
```

Speicherplatzprotokollierung bei ZFS-Speicherpool

Der SIZE-Wert, der von dem Befehl `zpool list` protokolliert wird, besteht im Allgemeinen aus dem physischen Festplattenspeicher in dem Pool, variiert jedoch je nach Redundanzebene des Pools. Hierzu wird auf die folgenden Beispiele verwiesen. Der Befehl `zfs list` führt den verwendbaren Speicherplatz auf, der für Dateisysteme verfügbar ist, d.h. der Festplattenspeicher minus dem Overhead der Redundanzmetadaten des ZFS-Pools, sofern vorhanden.

- **Nicht-redundanter Speicherpool** – Wenn ein Pool mit einer 136-GB-Festplatte erstellt wird, protokolliert der Befehl `zpool list` den Wert `SIZE` und den anfänglichen Wert `FREE` als 136 GB. Der anfängliche Speicherplatz `AVAIL`, der von dem Befehl `zfs list` protokolliert wird, beträgt 134 GB, aufgrund des kleinen Overheads durch die Poolmetadaten. Beispiel:

  ```
  # zpool create tank c0t6d0
  # zpool list tank
  NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
  tank   136G  95.5K   136G     0%  1.00x  ONLINE  -
  # zfs list tank
  NAME   USED  AVAIL  REFER  MOUNTPOINT
  tank    72K   134G    21K  /tank
  ```

- **Gespiegelter Speicherpool** – Wenn ein Pool mit zwei136-GB-Festplatten erstellt wird, protokolliert der Befehl `zpool list` den Wert `SIZE` als 136 GB und den anfänglichen Wert `FREE` als 136 GB. Diese Protokollierung wird als *verkleinerter* Speicherplatzwert bezeichnet. Der anfängliche Speicherplatz `AVAIL`, der von dem Befehl `zfs list` protokolliert wird, beträgt 134 GB, aufgrund des kleinen Overheads durch die Poolmetadaten. Beispiel:

  ```
  # zpool create tank mirror c0t6d0 c0t7d0
  # zpool list tank
  NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
  tank   136G  95.5K   136G     0%  1.00x  ONLINE  -
  # zfs list tank
  NAME   USED  AVAIL  REFER  MOUNTPOINT
  tank    72K   134G    21K  /tank
  ```

- **RAID-Z-Speicherpool** – Wenn ein `raidz2`-Pool mit drei 136-GB-Festplatten erstellt wird, protokolliert der Befehl `zpool list` den Wert `SIZE` als 408 GB und den anfänglichen Wert `FREE` als 408 GB. Diese Protokollierung wird als *vergrößerter* Festplattenspeicherwert bezeichnet, der Redundanz-Overhead umfasst, wie Paritätsinformationen. Der anfängliche Speicherplatz `AVAIL` , der von dem Befehl `zfs list` protokolliert wird, beträgt 133 GB

wegen des Poolredundanz-Overheads. Die Speicherplatzdiskrepanz zwischen der Ausgabe des Befehls `zpool list` und des Befehls `zfs list` für einen RAID-Z-Pool ist darauf zurückzuführen, dass `zpool list` den vergrößerten Poolspeicherplatz protokolliert.

```
# zpool create tank raidz2 c0t6d0 c0t7d0 c0t8d0
# zpool list tank
NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
tank   408G   286K   408G     0%  1.00x  ONLINE  -
# zfs list tank
NAME   USED  AVAIL  REFER  MOUNTPOINT
tank  73.2K   133G  20.9K  /tank
```

Reparieren beschädigter Daten

In den folgenden Abschnitten wird beschrieben, wie Sie den Typ der Datenbeschädigung ermitteln und (falls möglich) Daten reparieren können.

- „Ermitteln der Art der Datenbeschädigung“ auf Seite 324
- „Reparatur beschädigter Dateien bzw. Verzeichnisse“ auf Seite 325
- „Reparieren von Schäden am gesamten ZFS-Speicher-Pool“ auf Seite 326

Zur Minimierung des Risikos von Datenbeschädigung nutzt ZFS Prüfsummenberechnung, Redundanz und Datenselbstheilung. Dennoch können Datenbeschädigungen auftreten, wenn ein Pool nicht redundant ist. Dies ist der Fall, wenn sich der Pool zum Zeitpunkt der Beschädigung in beeinträchtigtem Zustand befindet oder mehrere Datenkopien durch unvorhergesehene Ereignisse beschädigt werden. Unabhängig von der Ursache ist das Ergebnis dasselbe: Daten werden beschädigt und sind deswegen nicht mehr verfügbar. Die erforderlichen Abhilfemaßnahmen hängen von der Art der beschädigten Daten und ihrem relativen Wert ab. Es können zwei grundlegende Arten von Daten beschädigt werden:

- Pool-Metadaten – Damit Pools geöffnet werden können und auf Datasets zugegriffen werden kann, muss ZFS eine Reihe spezieller Daten verarbeiten. Wenn diese Daten beschädigt sind, stehen der gesamte Pool bzw. ein Teil der Datasets nicht mehr zur Verfügung.
- Objektdaten – In diesem Fall sind Daten innerhalb einer bestimmten Datei bzw. eines Verzeichnisses beschädigt. Dieses Problem kann dazu führen, dass auf einen Teil dieser Datei bzw. dieses Verzeichnisses nicht mehr zugegriffen werden kann oder das betreffende Objekt in seiner Gesamtheit nicht mehr verfügbar ist.

Daten werden während des Normalbetriebs und der Datenbereinigung auf Integrität überprüft. Weitere Informationen zum Überprüfen der Integrität von Pool-Daten finden Sie unter „Überprüfen der Integrität des ZFS-Dateisystems“ auf Seite 318.

Ermitteln der Art der Datenbeschädigung

Der Befehl `zpool status` zeigt standardmäßig nur an, dass eine Datenbeschädigung aufgetreten ist, es ist jedoch nicht ersichtlich, wo sie sich ereignet hat. Beispiel:

```
# 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
```

Ein Fehler zeigt lediglich an, dass zu einem bestimmten Zeitpunkt ein Fehler aufgetreten ist. Diese Fehler müssen nicht mehr notwendigerweise im System vorhanden sein. Unter normalen Bedingungen ist dies der Fall. Bestimmte zeitweilige Ausfälle können zu Datenbeschädigungen führen, die nach dem Ende des Ausfalls automatisch behoben werden. Bei einer vollständigen Bereinigung des Pools wird jeder aktive Datenblock im Pool untersucht, und das Fehlerprotokoll wird nach Abschluss der Bereinigung geleert. Wenn Sie sehen, dass die betreffenden Fehler im Pool nicht mehr auftreten und Sie nicht auf den Abschluss des Bereinigungsvorgangs warten möchten, können Sie alle Fehler im Pool mithilfe des Befehls `zpool online` löschen.

Wenn die Datenbeschädigung in Metadaten für den gesamten Pool aufgetreten ist, sieht die Befehlsausgabe etwas anders aus. Beispiel:

```
# 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
```

Ist ein Pool beschädigt, wird er in den Status `FAULTED` versetzt, da er nicht die erforderliche Datenredundanz gewährleisten kann.

Reparatur beschädigter Dateien bzw. Verzeichnisse

Ist eine Datei oder ein Verzeichnis beschädigt, kann es je nach Art der Datenbeschädigung sein, dass das System noch immer funktioniert. Schäden sind praktisch nicht wieder rückgängig zu machen, wenn im System keine brauchbaren Datenkopien vorhanden sind. Wenn die betreffenden Daten wertvoll sind, müssen Sie diese aus Sicherungskopien wiederherstellen. Auch in diesem Fall kann es sein, dass Sie nach dieser Datenbeschädigung den Normalbetrieb wiederherstellen können, ohne das gesamte Pool wiederherstellen zu müssen.

Wenn sich der Schaden innerhalb eines Dateidatenblocks befindet, kann die betreffende Datei sicher gelöscht werden, wodurch der Fehler aus dem System entfernt wird. Mit dem Befehl `zpool status -v` können Sie eine Liste von Dateinamen mit dauerhaften Fehlern ausgeben. Beispiel:

```
# 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
```

Es folgen Erläuterung zur Liste von Dateinamen mit dauerhaften Fehlern:

- Wenn der vollständige Pfad zur Datei gefunden wurde und das Dataset eingehängt ist, wird der vollständige Pfad zur Datei angezeigt. Beispiel:

  ```
  /monkey/a.txt
  ```

- Wenn der vollständige Pfad zur Datei gefunden wurde, das Dataset jedoch nicht eingehängt ist, wird der Dataset-Name ohne vorangestellten Schrägstrich (/), gefolgt vom Pfad zur Datei innerhalb des Datasets angezeigt. Beispiel:

  ```
  monkey/ghost/e.txt
  ```

- Wenn die Objektnummer zu einem Dateipfad nicht erfolgreich konvertiert werden kann, weil ein Fehler auftrat oder dem Objekt kein realer Pfad zugewiesen ist (z. B. bei `dnode_t`), dann wird der Dataset-Name, gefolgt von der Objektnummer angezeigt. Beispiel:

```
monkey/dnode:<0x0>
```

- Wenn ein Objekt im Metaobjekt-Set (MOS) beschädigt ist, wird ein spezielles Tag `<metadata>`, gefolgt von der Objektnummer, angezeigt.

Wenn Metadaten einer Datei bzw. eines Verzeichnisses beschädigt sind, kann die betreffende Datei nur an einen anderen Ort verschoben werden. Sie können Dateien bzw. Verzeichnisse sicher an eine unkritische Stelle kopieren, sodass das ursprüngliche Objekt am Ausgangsort wiederhergestellt werden kann.

Abhilfe bei beschädigten Daten mit mehreren Blockreferenzen

Wenn ein beschädigtes Dateisystem fehlerhafte Daten mit mehreren Blockreferenzen enthält, wie Schnappschüssen, kann der Befehl `zpool status -v` nicht **alle** fehlerhaften Datenpfade anzeigen. Die aktuelle `zpool status`-Protokollierung der beschädigten Daten wird durch den Umfang der Metadatenbeschädigung und außerdem dadurch begrenzt, ob Blöcke nach Ausführung des Befehls `zpool status` wieder verwendet wurden. Deduplizierte Blöcke machen die Protokollierung aller fehlerhaften Daten noch komplizierter.

Wenn fehlerhafte Daten vorhanden sind und der Befehl `zpool status -v` identifiziert, dass Schnappschussdaten betroffen sind, sollten Sie den folgenden Befehl ausführen, um zusätzliche fehlerhafte Pfade zu identifizieren:

Reparieren von Schäden am gesamten ZFS-Speicher-Pool

Wenn Pool-Metadaten beschädigt sind und der Pool dadurch nicht geöffnet oder importiert werden kann, stehen Ihnen folgende Optionen zur Verfügung:

- Sie können versuchen, den Pool mithilfe des Befehls `zpool clear -F` oder `zpool import -F` wiederherzustellen. Mithilfe dieser Befehle wird versucht, die letzten Pool-Transaktionen zurückzusetzen, um den Pool wiederherzustellen. Sie können den Befehl `zpool status` verwenden, um einen beschädigten Pool und die empfohlenen Wiederherstellungsschritte anzuzeigen. Beispiel:

```
# 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
```

Für den in der vorhergehenden Ausgabe beschriebenen Wiederherstellungsvorgang wird folgender Befehl ausgeführt:

```
# zpool clear -F tpool
```

Wenn Sie versuchen, einen beschädigten Pool zu importieren, wird eine Meldung wie die folgende angezeigt:

```
# 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.
```

Für den in der vorhergehenden Ausgabe beschriebenen Wiederherstellungsvorgang wird folgender Befehl ausgeführt:

```
# 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
```

Wenn der beschädigte Pool in der Datei `zpool.cache` enthalten ist, wird das Problem behoben, sobald das System neu gestartet wird, und die Beschädigung des Pools wird über den Befehl `zpool status` angezeigt. Falls der Pool nicht in der Datei `zpool.cache` enthalten ist, kann er nicht importiert oder geöffnet werden, und es werden Meldungen angezeigt, die den beschädigten Pool betreffen, wenn Sie versuchen, den Pool zu importieren.

- Sie können einen beschädigten Pool im schreibgeschützten Modus importieren. Mit dieser Methode können Sie den Pool so importieren, dass Sie Zugriff auf die Daten haben. Beispiel:

  ```
  # zpool import -o readonly=on tpool
  ```

  Weitere Informationen zum schreibgeschützten Importieren eines Pools finden Sie unter „Importieren eines Pools im schreibgeschützten Modus“ auf Seite 106.

- Mit dem Befehl `zpool import -m` können Sie einen Pool mit einem fehlenden Protokolliergerät importieren. Weitere Informationen finden Sie unter „Importieren eines Pools mit fehlendem Protokolliergerät“ auf Seite 105.
- Wenn der Pool durch keine der beiden Wiederherstellungsmethoden wiederhergestellt werden kann, müssen Sie den Pool und alle seine Daten aus einer Sicherungskopie wiederherstellen. Das eingesetzte Verfahren hängt weitgehend von der Pool-Konfiguration und der Datensicherungsstrategie ab. Zuerst sollten Sie die mit dem Befehl `zpool status` angezeigte Konfiguration speichern, um diese nach dem Löschen des Pools wiederherstellen zu können. Löschen Sie den Pool dann mit dem Befehl `zpool destroy -f`.

Legen Sie sich an sicherer Stelle eine Datei an, in der die Struktur der Datasets sowie die einzelnen lokal gesetzten Eigenschaften beschrieben sind, da diese Informationen nicht mehr zugänglich sind, wenn auf den Pool nicht mehr zugegriffen werden kann. Mithilfe der Pool-Konfiguration und der Dataset-Struktur können Sie nach dem Löschen des Pools die vollständige Konfiguration wiederherstellen. Danach können Sie mithilfe eines beliebigen Wiederherstellungsverfahrens den Pool wieder mit Daten „auffüllen“.

Reparieren einer beschädigten ZFS-Konfiguration

ZFS unterhält im Root-Dateisystem einen Cache aktiver Pools und ihrer Konfiguration. Wenn diese Cache-Datei beschädigt wird bzw. nicht mehr mit den auf dem Datenträger gespeicherten Konfigurationsinformationen übereinstimmt, kann auf einen Pool nicht mehr zugegriffen werden. ZFS versucht, eine solche Situation zu vermeiden, obwohl aufgrund der Eigenschaften des zugrunde liegenden Speichers immer die Möglichkeit besteht, dass Daten beschädigt werden können. Solche Situationen führen normalerweise zu einem Verschwinden eines ansonsten verfügbaren Pools aus dem System und können sich auch in unvollständigen Konfigurationen manifestieren, denen eine unbekannte Anzahl an virtuellen Geräten der obersten Hierarchie fehlt. In allen Fällen kann die Konfiguration durch Exportieren des Pools (falls er zugänglich ist) und anschließendes Importieren wiederhergestellt werden.

Weitere Informationen zum Importieren und Exportieren von Pools finden Sie unter „Migrieren von ZFS-Speicher-Pools“ auf Seite 100.

Reparieren eines Systems, das nicht hochgefahren werden kann

ZFS soll trotz möglicher Fehler robust und stabil sein. Dennoch können Softwarebug oder bestimmte unerwartete Probleme zum Systemabsturz führen, wenn auf einen Pool zugegriffen wird. Jeder Pool muss im Rahmen des Systemstarts geöffnet werden, was bedeutet, dass Fehler beim Zugreifen auf einen Pool in eine Systemabsturz-Neustart-Schleife münden können. Als Ausweg aus einer solchen Situation muss ZFS mitgeteilt werden, Pools beim Systemstart zu ignorieren.

ZFS unterhält in `/etc/zfs/zpool.cache` einen internen Cache aktiver Pools und ihrer Konfiguration. Der Speicherort und der Inhalt dieser Datei sind nicht öffentlich zugänglich und Änderungen unterworfen. Wenn ein System nicht mehr hochgefahren werden kann, sollten Sie es mit der Meilenstein-Option `none` (`-m milestone=none`) booten. Nach dem Hochfahren des Systems hängen Sie das Root-Dateisystem ohne Schreibschutz ein. Anschließend benennen Sie die Datei `/etc/zfs/zpool.cache` um oder verschieben diese. Dadurch "vergisst" ZFS, dass im System Pools vorhanden sind und greift nicht auf den schadhaften Pool zu, der das Problem verursacht hat. Danach können Sie durch Absetzen des Befehls `svcadm milestone all` den

normalen Systemzustand wiederherstellen. Beim Booten von einem alternativen Root-Verzeichnis zu Reparaturzwecken können Sie ein ähnliches Verfahren anwenden.

Wenn das System wieder läuft, können Sie versuchen, den Pool mithilfe des Befehls `zpool import` zu importieren. Dadurch tritt jedoch wahrscheinlich der gleiche Fehler wie beim Systemstart auf, da dieser Befehl zum Zugriff auf Pools das gleiche Verfahren verwendet. Wenn das System mehrere Pools besitzt, gehen Sie wie folgt vor:

- Benennen Sie die Datei `zpool.cache` um oder verschieben Sie sie in ein anderes Verzeichnis (siehe oben).
- Finden Sie heraus, bei welchem Pool Probleme auftreten, indem Sie sich mithilfe des Befehls `fmdump -eV` die Pools, für die kritische Fehler gemeldet wurden, anzeigen lassen.
- Importieren Sie die Pools nacheinander. Lassen Sie dabei die Pools aus, bei denen Probleme aufgetreten sind, die in der Ausgabe des Befehls `fmdump` angezeigt werden.

◆ ◆ ◆ KAPITEL 11

Empfohlene Oracle Solaris ZFS-Vorgehensweisen

In diesem Kapitel werden die empfohlenen Vorgehensweisen zum Erstellen, Überwachen und Verwalten der ZFS-Speicherpools und Dateisysteme beschrieben.

Dieses Kapitel enthält die folgenden Abschnitte:

Empfohlene Vorgehensweise bei Speicherpools

Die folgenden Abschnitte enthalten empfohlene Vorgehensweise zum Erstellen und Überwachen von ZFS-Speicherpools. Weitere Informationen über die Behebung von Problemen mit Speicherpools finden Sie in Kapitel 10, „Problembehebung und Pool-Wiederherstellung in Oracle Solaris ZFS“.

Allgemeine Vorgehensweise bei dem System

- System mit den neuesten Solaris-Versionen und -Patches auf dem aktuellen Stand halten
- Prüfen Sie, ob der Controller Cache-Flush-Befehle akzeptiert, so dass die Daten sicher geschrieben werden. Dies ist wichtig, bevor Sie die Geräte des Pools ändern oder einen gespiegelten Speicherpool teilen. Dies ist im Allgemeinen bei Oracle/Sun-Hardware kein Problem, die Prüfung, ob die Cache-Flushing-Einstellung der Hardware aktiviert ist, wird jedoch in jedem Fall empfohlen.
- Passen Sie die Größe des Speichers der aktuellen System-Workload an.
  - Mit einem bekannten Speicherbedarf einer Anwendung, wie beispielsweise einer Datenbankanwendung, können Sie die ARC-Größe kappen, sodass die Anwendung den erforderlichen Speicher nicht aus dem ZFS-Cache anfordern muss.
  - Speicheranforderungen für Deduplizierung berücksichtigen

- Bestimmen Sie die ZFS-Speicherauslastung mit dem folgenden Befehl:

```
# mdb -k
> ::memstat
Page Summary                Pages                MB  %Tot
------------     ----------------  ----------------  ----
Kernel                     388117              1516   19%
ZFS File Data               81321               317    4%
Anon                        29928               116    1%
Exec and libs                1359                 5    0%
Page cache                   4890                19    0%
Free (cachelist)             6030                23    0%
Free (freelist)           1581183              6176   76%

Total                     2092828              8175
Physical                  2092827              8175
> $q
```

- Erwägen Sie die Verwendung von ECC-Speicher zum Schutz vor Speicherbeschädigung. Eine unbemerkte Speicherbeschädigung kann Ihre Daten potenziell beschädigen.

- Führen Sie regelmäßige Backups durch - Auch wenn ein Pool, der mit ZFS-Redundanz erstellt wird, Ausfallzeiten wegen Hardwarefehlern verringern kann, ist er nicht gegen Hardwarefehler, Stromausfälle oder getrennte Kabelverbindungen gefeit. Sie müssen Ihre Daten in regelmäßigen Abständen sichern. Wichtige Daten müssen gesichert werden. Kopien von Daten können auf unterschiedliche Weise erstellt werden:
  - Regelmäßige oder tägliche ZFS-Schnappschüsse
  - Wöchentliche Backups von ZFS-Pooldaten. Mit dem Befehl `zpool split` können Sie ein genaues Duplikat des gespiegelten ZFS-Speicherpools erstellen.
  - Monatliche Backups mit einem Backupprodukt auf Unternehmensebene
- Hardware-RAID
  - Sie sollten den JBOD-Modus für Speichermatrizes anstelle von Hardware-RAID verwenden, damit ZFS den Speicher und die Redundanz verwalten kann.
  - Hardware-RAID und/oder ZFS-Redundanz verwenden
  - Die Verwendung der ZFS-Redundanz bietet viele Vorteile - Konfigurieren Sie ZFS bei Production-Umgebungen so, dass Dateninkonsistenzen behoben werden. Verwenden Sie ZFS-Redundanz, wie RAID-Z, RAID-Z-2, RAID-Z-3, Spiegelung, ungeachtet der RAID-Ebene, die in dem zugrunde liegenden Speichergerät implementiert ist. Mit dieser Redundanz können Fehler im zugrunde liegenden Speichergerät oder bei dessen Verbindungen zu dem Host erkannt und von ZFS behoben werden.

  Weitere Informationen finden Sie in „Vorgehensweise beim Erstellen von Pools auf lokalen oder netzwerkgestützten Speichermatrizes“ auf Seite 335.
- Absturz-Dumps belegen mehr Plattenkapazität, im Allgemeinen in der Größenordnung von 1/2-3/4 des physischen Speicherbereichs.

Vorgehensweise beim Erstellen von ZFS-Speicherpools

Die folgenden Abschnitte enthalten allgemeine und spezifischere Vorgehensweisen im Zusammenhang mit Pools

Allgemeine Vorgehensweise bei Speicherpools

- Verwenden Sie ganze Festplatten, um den Festplattenschreibungscache zu aktivieren und einfachere Wartung zu ermöglichen. Das Erstellen von Pools in Bereichen macht die Verwaltung und Wiederherstellung von Festplatten komplexer.
- Verwenden Sie ZFS-Redundanz, damit ZFS Dateninkonsistenzen beheben kann.
  - Die folgende Meldung wird angezeigt, wenn ein nicht-redundanter Pool erstellt wird:

    ```
    # zpool create tank c4t1d0 c4t3d0
    'tank' successfully created, but with no redundancy; failure
    of one device will cause loss of the pool
    ```
  - Verwenden Sie gespiegelte Festplattenpaare für gespiegelte Pools
  - Gruppieren Sie 3-9 Festplatten pro VDEV bei RAID-Z-Pools.
  - Mischen Sie RAID-Z und gespiegelte Komponenten nicht innerhalb desselben Pools. Diese Pools sind schwieriger zu verwalten und die Leistung kann beeinträchtigt werden.
- Verwenden Sie Hot-Spares, um die Ausfallzeit aufgrund von Hardwarefehlern zu reduzieren.
- Verwenden Sie Festplatten mit ähnlicher Größe, damit die E/A über die Geräte hinweg ausgeglichen ist.
  - Kleinere LUNs können zu großen LUNs erweitert werden
  - Erweitern Sie keine LUNs auf extrem unterschiedliche Größen, wie von 128 MB auf 2 TB, um optimale Metaslab-Größen beizubehalten.
- Erwägen Sie das Erstellen eines kleinen Root-Pools und größerer Datenpools, um eine schnellere Systemwiederherstellung zu unterstützen.
- Die empfohlene minimale Poolgröße beträgt 8 GB. Auch wenn die minimale Poolgröße 64 MB beträgt, macht jede Größe unter 8 GB die Zuweisung und Anforderung von freier Poolkapazität schwieriger.
- Die empfohlene maximale Poolgröße muss Ihre Workload- oder Datengröße bequem bewältigen können. Versuchen Sie nicht, mehr Daten zu speichern, als Sie routinemäßig regelmäßig sichern können. Sonst werden Ihre Daten unvorhergesehenen Ereignissen ausgesetzt.

Vorgehensweise beim Erstellen von Root-Pools

- Erstellen Sie Root-Pools mit Bereichen, indem Sie den Bezeichner `s*` verwenden. Verwenden Sie die Bezeichnung `p*` nicht. Im Allgemeinen wird der ZFS-Root-Pool eines Systems bei der Installation des Systems erstellt. Wenn Sie einen zweiten Root-Pool erstellen oder einen Root-Pool erneut erstellen, verwenden Sie eine Syntax wie die Folgende:

  ```
  # zpool create rpool c0t1d0s0
  ```

  Oder erstellen Sie einen gespiegelten Root-Pool. Beispiel:

  ```
  # zpool create rpool mirror c0t1d0s0 c0t2d0s0
  ```

  - Der Root-Pool muss als gespiegelte Konfiguration oder als einzelne Festplattenkonfiguration erstellt werden. Weder eine RAID-Z- noch eine Stripeset-Konfiguration wird unterstützt. Sie können mit dem Befehl `zpooladd` keine zusätzlichen Festplatten hinzufügen, um mehrere gespiegelte virtuelle Geräte der obersten Hierarchieebene zu erstellen, aber Sie können ein gespiegeltes virtuelles Gerät mit dem Befehl `zpool attach` erweitern.
  - Der Root-Pool kann über kein separates Protokolliergerät verfügen.
  - Pooleigenschaften können während einer AI-Installation festgelegt werden, der `gzip`-Komprimierungsalgorithmus wird bei Root-Pools jedoch nicht unterstützt.
  - Benennen Sie den Root-Pool nicht um, nachdem dieser in einer Erstinstallation erstellt wurde. Das Umbenennen des Root-Pools kann dazu führen, dass das System nicht gebootet werden kann.
  - Erstellen Sie keinen Root-Pool auf einem USB-Stick bei einem Production-System, weil Root-Poolfestplatten für den fortlaufenden Betrieb wichtig sind, insbesondere in einer Unternehmensumgebung. Sie sollten die internen Festplatten eines Systems für den Root-Pool verwenden oder mindestens Festplatten derselben Qualität, die Sie für Nicht-Root-Daten verwenden. Außerdem ist ein USB-Stick möglicherweise nicht groß genug zur Aufnahme einer Dump-Volume-Größe, die mindestens der Hälfte der Größe des physischen Speichers entspricht.

Vorgehensweise beim Erstellen von Nicht-Root-Pools

- Erstellen Sie Nicht-Root-Pools mit ganzen Festplatten mit dem Bezeichner `d*`. Verwenden Sie die Bezeichnung `p*` nicht.
  - ZFS funktioniert am besten ohne zusätzliche Software zur Datenträgerverwaltung.
  - Für bessere Leistung verwenden Sie einzelne Festplatten oder mindestens LUNs, die nur aus wenigen Festplatten bestehen. Wenn ZFS mehr Einblick in das LUNs-Setup hat, kann ZFS bessere E/A-Planungsentscheidungen treffen.
  - Erstellen Sie redundante Poolkonfigurationen über mehrere Controller hinweg, um die Ausfallzeiten aufgrund eines Controller-Fehlers zu reduzieren.
  - **Gespiegelte Speicherpools** – Belegen weniger Plattenkapazität, die Leistung ist jedoch im Allgemeinen bei kleinen zufälligen Lesezugriffen besser.

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

- **RAID-Z-Speicherpools** – können mit 3 Paritätsstrategien erstellt werden, wobei Parität gleich 1 (`raidz`), 2 (`raidz2`) oder 3 (`raidz3`) ist. Ein RAID-Z-Konfiguration maximiert die Plattenkapazität und erbringt im Allgemeinen eine gute Leistung, wenn Daten in großen Blöcken (128 K oder mehr) geschrieben werden.
  - Sie sollten eine RAID-Z-(`raidz-`)Konfiguration mit einfacher Parität mit 2 VDEVs aus jeweils 3 Festplatten (2+1) in Erwägung ziehen.

    ```
    # zpool create rzpool raidz1 c1t0d0 c2t0d0 c3t0d0 raidz1 c1t1d0 c2t1d0 c3t1d0
    ```
  - Eine RAIDZ-2-Konfiguration bietet bessere Datenverfügbarkeit und die Leistung entspricht in etwa RAID-Z. RAIDZ-2 hat eine wesentlich bessere Mean Time to Data Loss (MTTDL) als RAID-Z oder 2-Way-Spiegelung. Erstellen Sie eine RAID-Z-(`raidz2-`)Konfiguration mit doppelter Parität auf 6 Festplatten (4+2).

    ```
    # zpool create rzpool raidz2 c0t1d0 c1t1d0 c4t1d0 c5t1d0 c6t1d0 c7t1d0
    raidz2 c0t2d0 c1t2d0 c4t2d0 c5t2d0 c6t2d0 c7t2d
    ```
  - Eine RAID-3-Konfiguration maximiert die Plattenkapazität und bietet optimale Verfügbarkeit, weil sie 3 Festplattenfehlern auffangen kann. Erstellen Sie eine RAID-Z-(`raidz3-`)Konfiguration mit dreifacher Parität auf 9 Festplatten (6+3).

    ```
    # zpool create rzpool raidz3 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0
    c5t0d0 c6t0d0 c7t0d0 c8t0d0
    ```

Vorgehensweise beim Erstellen von Pools auf lokalen oder netzwerkgestützten Speichermatrizes

Beachten Sie die folgende Vorgehensweise beim Erstellen eines ZFS-Speicherpools in einer Speichermatrix, die lokal oder entfernt verbunden ist.

- Wenn Sie einen Pool auf SAN-Geräten erstellen und die Netzwerkverbindung langsam ist, befinden sich die Geräte des Pools während einer bestimmten Zeit im Status `UNAVAIL`. Sie müssen testen, ob die Netzwerkverbindung Ihre Daten fortlaufend bereitstellen kann. Beachten Sie außerdem bei der Verwendung von SAN-Geräten für Ihren Root-Pool, dass die Geräte möglicherweise nicht verfügbar sind, wenn das System gebootet wird, und dass sich die Geräte des Root-Pools auch im Status `UNAVAIL` befinden können.
- Stellen Sie mit dem Hersteller der Matrix sicher, dass die Festplattenmatrix den Cache nicht leert, nachdem eine "Flush Write Cache"-Anforderung von ZFS ausgegeben wird.
- Verwenden Sie ganze Festplatten und keine Festplattenbereiche als Speicherpoolgeräte, damit Oracle Solaris ZFS die lokalen kleinen Festplattencaches aktiviert, die zu gegebener Zeit geleert werden.
- Für optimale Leistung erstellen Sie eine LUN für jede physische Festplatte in der Matrix. Die Verwendung von nur einer großen LUN kann dazu führen, dass ZFS zu wenig E/A-Vorgänge für eine optimale Speicherung in die Warteschlange stellt. Umgekehrt kann die Verwendung von vielen kleinen LUNs dazu führen, dass der Speicher mit einer großen Anzahl von ausstehenden Lese-E/A-Vorgängen überschwemmt wird.

- Eine Speichermatrix, die dynamische (oder schlanke) Provisioning-Software zur Implementierung der virtuellen Speicherplatzzuweisung verwendet, wird für Oracle Solaris ZFS nicht empfohlen. Wenn Oracle Solaris ZFS die geänderten Daten in freien Speicherplatz schreibt, wird in die ganze LUN geschrieben. Der Oracle Solaris ZFS-Schreibprozess weist den virtuellen Speicherplatz aus Sicht der Speichermatrix zu, wodurch der Vorteil des dynamischen Provisionings zunichte gemacht wird.

  Eine dynamische Provisioning-Software ist also bei Verwendung von ZFS nicht unbedingt erforderlich.

  - Sie können eine LUN in einem vorhandenen ZFS-Speicher erweitern, sodass der neue Speicherplatz verwendet wird.
  - Dasselbe Verhalten gilt, wenn eine kleinere LUN durch eine größere LUN ersetzt wird.
  - Wenn Sie den Speicherplatzbedarf für Ihren Pool testen und den Pool mit kleineren LUNS erstellen, die die Speicherplatzanforderungen erfüllen, können Sie die LUNs jederzeit vergrößern, wenn Sie mehr Platz benötigen.
- Wenn die Matrix einzelne Geräte enthalten kann (JBOD-Modus), sollten Sie redundante ZFS-Speicherpools (Spiegelung oder RAID-Z) bei diesem Matrixtyp erstellen, damit ZFS Dateninkonsistenzen protokollieren und beheben kann.

Vorgehensweise beim Erstellen eines Pools für eine Oracle-Datenbank

Beim Erstellen einer Oracle-Datenbank sollten Sie folgende Vorgehensweise bei Speicherpools beachten.

- Verwenden Sie einen gespiegelten Pool oder Hardware-RAID für Pools
- RAID-Z-Pools werden generell für Workloads mit zufälligen Lesezugriffen nicht empfohlen
- Erstellen Sie einen kleinen separaten Pool mit einem separaten Loggerät für Datenbank-Redo-Logs.
- Erstellen Sie einen kleinen separaten Pool für das Archive-Log.

Weitere Informationen finden Sie im folgenden White Paper.

`http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle`

Verwenden von ZFS-Speicherpools in VirtualBox

- VirtualBox ist so konfiguriert, dass Cache-Flush-Befehle aus dem zugrunde liegenden Speicher standardmäßig ignoriert werden. Dies bedeutet, dass Daten bei einem Systemabsturz oder Hardwarefehler verloren gehen können.
- Aktivieren Sie das Cache-Flushing in VirtualBox durch Ausgabe des folgenden Befehls:

```
VBoxManage setextradata <VM_NAME> "VBoxInternal/Devices/<type>/0/LUN#<n>/Config/IgnoreFlush" 0
```

  - <VM_NAME> ist der Name des virtuellen Rechners

- <type> ist der Controller-Typ, entweder `piix3ide` (wenn Sie den üblichen virtuellen IDE-Controller verwenden) oder `ahci`, wenn Sie einen SATA-Controller verwenden.
- <n> ist die Festplattennummer

Vorgehensweise bei Speicherpools im Hinblick auf Leistung

- Halten Sie die Poolkapazität im Hinblick auf optimale Leistung unter 80 %
- Gespiegelte Pools werden vor RAID-Z-Pools bei Workloads mit zufälligen Lese-/Schreibvorgängen empfohlen.
- Separate Loggeräte
  - Werden zur Verbesserung der Leistung bei synchronen Schreibvorgängen empfohlen.
  - Bei einer hohen synchronen Schreibbelastung wird die Fragmentierung durch das Schreiben vieler Logblöcke in den Hauptpool vermieden.
- Separate Cachegeräte werden empfohlen, um die Leseleistung zu verbessern.
- Bereinigen/Wiederherstellen - Bei einem sehr großen RAID-Z-Pool mit vielen Geräten kommt es zu längeren Bereinigungs- und Wiederherstellungszeiten
- Poolleistung ist langsam – Verwenden Sie den Befehl `zpool status`, um Hardwareprobleme zu ermitteln, die zu Problemen mit der Poolleistung führen. Wenn in dem Befehl `zpool status` keine Probleme aufgeführt werden, verwenden Sie den Befehl `fmdump` zur Anzeige von Hardwarefehlern, oder verwenden Sie den Befehl `fmdump -eV`, um Hardwarefehler zu prüfen, die noch zu keiner Fehlermeldung geführt haben.

Vorgehensweise bei Verwaltung und Überwachung des ZFS-Speicherpools

- Stellen Sie die sicher, dass die Poolkapazität unter 80 % liegt, um eine optimale Leistung zu erzielen.

  Die Poolleistung kann sich verringern, wenn ein Pool sehr voll ist und Dateisysteme häufig aktualisiert werden, wie bei einem stark beanspruchten Mailserver. Volle Pools können zu Leistungseinbußen führen, jedoch zu keinen anderen Problemen. Wenn die primäre Workload aus unveränderlichen Dateien besteht, halten Sie den Pool im Auslastungsbereich von 95-96 %. Selbst bei im Wesentlichen statischem Inhalt kann sich die Lese-, Schreib- und Wiederherstellungsleistung im Bereich von 95-96 % verringern.
  - Überwachen Sie den Speicherplatz im Pool und Dateisystem, um sicherzustellen, dass diese nicht voll sind.
  - Sie sollten die Verwendung von ZFS-Quota und Reservierungen in Erwägung ziehen, um sicherzustellen, dass das Dateisystem nicht mehr als 80 % der Poolkapazität belegt.

- Überwachen der Poolintegrität
  - Redundante Pools - Überwachen Sie den Pool mit den Befehlen `zpool status` und `fmdump` wöchentlich
  - Nicht-redundante Pools - Überwachen Sie den Pool mit den Befehlen `zpool status` und `fmdump` alle zwei Wochen
- Führen Sie den Befehl `zpool scrub` regelmäßig aus, um Probleme mit der Datenintegrität zu identifizieren.
  - Bei Laufwerken mit Verbraucherqualität sollten Sie einen wöchentlichen Bereinigungsplan festlegen.
  - Bei Laufwerken mit DataCenter-Qualität sollten Sie einen monatlichen Bereinigungsplan festlegen.
  - Außerdem sollten Sie eine Bereinigung durchführen, bevor Sie Geräte ersetzen oder die Redundanz eines Pools vorübergehend verringern, um sicherzustellen, dass alle Geräte aktuell einsatzfähig sind.
- Überwachen von Pool- oder Gerätefehlern - Verwenden Sie `zpool status` wie unten beschrieben. Verwenden Sie außerdem den Befehl `fmdump` oder `fmdump -eV`, um festzustellen, ob Geräte-Faults oder -Fehler aufgetreten sind.
  - Redundante Pools - Überwachen Sie die Poolintegrität mit den Befehlen `zpool status` und `fmdump` wöchentlich
  - Nicht-redundante Pools - Überwachen Sie die Poolintegrität mit den Befehlen `zpool status` und `fmdump` alle zwei Wochen
- Poolgerät ist `UNAVAIL` oder `OFFLINE` – Wenn ein Poolgerät nicht verfügbar ist, prüfen Sie, ob das Gerät in der Befehlsausgabe von `format` aufgeführt wird. Wenn das Gerät nicht in der `format`-Ausgabe aufgeführt wird, ist es für ZFS nicht sichtbar.

  Wenn ein Poolgerät den Status `UNAVAIL` oder `OFFLINE` hat, bedeutet dies im Allgemeinen, dass das Gerät fehlerhaft ist, dass die Kabelverbindung unterbrochen ist oder dass ein anderes Hardwareproblem vorliegt, wie ein fehlerhaftes Kabel oder ein fehlerhafter Controller, das dazu führt, dass nicht auf das Gerät zugegriffen werden kann.
- Überwachen des Speicherpoolplatzes – Verwenden Sie die Befehle `zpool list` und `zfs list`, um zu ermitteln, welcher Teil der Festplatte von Dateisystemdaten belegt wird. ZFS-Schnappschüsse können Plattenkapazität belegen. Wenn sie nicht von dem Befehl `zfs list` aufgeführt werden, können sie unbemerkt Datenträgerkapazität belegen. Verwenden Sie den Schnappschussbefehl `zfs list -t`, um Datenträgerkapazität zu ermitteln, die von Schnappschüssen belegt wird.

Empfohlene Vorgehensweise bei Dateisystemen

In den folgenden Abschnitten wird die empfohlene Vorgehensweise bei Dateisystemen beschrieben.

Vorgehensweise beim Erstellen von Dateisystemen

In den folgenden Abschnitten wird die Vorgehensweise beim Erstellen von ZFS-Dateisystemen beschrieben.

- Erstellen Sie ein Dateisystem pro Benutzer bei Home-Verzeichnissen
- Sie sollten die Verwendung von Dateisystemquota und -reservierungen zur Verwaltung und Reservierung von Plattenkapazität bei großen Dateisystemen in Erwägung ziehen.
- Sie sollten die Verwendung von Benutzer- und Gruppenquota zur Verwaltung der Plattenkapazität in einer Umgebung mit vielen Benutzern in Erwägung ziehen.
- Verwenden Sie die Vererbung der ZFS-Eigenschaft, um Eigenschaften für viele untergeordnete Dateisysteme anzuwenden.

Vorgehensweise beim Erstellen eines Dateisystems für eine Oracle-Datenbank

Beim Erstellen einer Oracle-Datenbank sollten Sie folgende Vorgehensweise bei Dateisystemen beachten.

- Gleichen Sie die ZFS-Eigenschaft `recordsize` mit der Oracle-Eigenschaft `db_block_size` ab.
- Erstellen Sie Datenbanktabellen- und Indexdateisysteme im Hauptdatenbankpool mit einem 8 KB-Wert für `recordsize` und dem Standardwert für `primarycache` .
- Erstellen Sie temporäre Daten- und Undo Tablespace-Dateisysteme im Hauptdatenbankpool mit den Standardwerten `recordsize` und `primarycache` .
- Erstellen Sie ein Archive-Log-Dateisystem im Archivpool. Dadurch werden die Komprimierung aktiviert und die Standardwerte `recordsize` und `primarycache` auf `metadata` gesetzt.

Weitere Informationen finden Sie im folgenden White Paper.

`http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle`

Vorgehensweise beim Überwachen von ZFS-Dateisystemen

Sie müssen die ZFS-Dateisysteme überwachen, um sicherzustellen, dass sie verfügbar sind und um Probleme bei der Speicherplatzbelegung zu identifizieren.

- Überwachen Sie die Verfügbarkeit des Dateisystemspeicherplatzes mit den Befehlen `zpool list` und `zfs list` und nicht mit den Befehlen `du` und `df`, weil Legacy-Befehle Speicherplatz nicht berücksichtigen, der von untergeordneten Dateisystemen oder Schnappschüssen belegt wird.

  Weitere Informationen finden Sie unter „Abhilfe bei ZFS-Speicherplatzproblemen“ auf Seite 321.
- Zeigen Sie die Speicherplatzbelegung des Dateisystems mit dem Befehl `zfs list -o space` an.
- Speicherplatz des Dateisystems kann unbemerkt von Schnappschüssen belegt werden. Sie können alle Dataset-Informationen mit folgender Syntax anzeigen:

  ```
  # zfs list -t all
  ```
- Ein separates `/var` -Dateisystem wird automatisch erstellt, wenn ein System installiert wird. Sie müssen jedoch die Quota und Reservierung für dieses Dateisystem festlegen, um sicherzustellen, dass nicht unbemerkt Speicherplatz des Root-Pools belegt wird.
- Außerdem können Sie die Dateivorgangsaktivität der ZFS-Dateisysteme mit dem Befehl `fsstat` anzeigen. Aktivitäten können nach Einhängepunkt oder Dateisystemtypen protokolliert werden. Das folgende Beispiel zeigt eine allgemeine Aktivität eines ZFS-Dateisystems:

  ```
  # fsstat /
   new  name   name  attr  attr lookup rddir  read read  write write
   file remov  chng   get   set    ops   ops   ops bytes   ops bytes
    832   589   286  837K 3.23K  2.62M 20.8K 1.15M 1.75G 62.5K  348M /
  ```
- Backups
  - Dateisystemschnappschüsse beibehalten
  - Sie sollten Software auf Unternehmensebene für wöchentliche und monatliche Backups verwenden.
  - Speichern Sie Schnappschüsse des Root-Pools auf einem entfernten System im Hinblick auf Bare Metal-Wiederherstellung

◆ ◆ ◆ ANHANG A

Oracle Solaris ZFS-Versionsbeschreibungen

In diesem Anhang werden die verfügbaren ZFS-Versionen und deren Funktionen beschrieben, sowie das Solaris-Betriebssystem, das die jeweilige ZFS-Version und die Funktionen bereitstellt.

Dieser Anhang enthält folgende Abschnitte:

Überblick über ZFS-Versionen

Mit einer speziellen ZFS-Version, die in den Solaris-Versionen enthalten ist, werden neue ZFS-Pool- und Dateisystemfunktionen bereitgestellt. Mit dem Befehl `zpool upgrade` oder `zfs upgrade` kann festgestellt werden, ob die Version, zu der ein Pool oder Dateisystem gehört, niedriger als die gerade laufende Solaris-Version ist. Mit diesen Befehlen können Sie auch Ihre Pool- und Dateisystemversionen aktualisieren.

Informationen zu den Befehlen `zpool upgrade` und `zfs upgrade` finden Sie unter „Aktualisieren von ZFS-Dateisystemen“ auf Seite 216 and „Aktualisieren von ZFS-Speicher-Pools“ auf Seite 109.

ZFS-Poolversionen

In der folgenden Tabelle sind die ZFS-Poolversionen aufgelistet, die in der Oracle Solaris-Version zur Verfügung stehen.

Version	Solaris 10	Beschreibung
1	Solaris 10 6/06	Ursprüngliche ZFS-Version
2	Solaris 10 11/06	Dito-Blöcke (replizierte Metadaten)
3	Solaris 10 11/06	Hot-Spares und RAID-Z mit doppelter Parität
4	Solaris 10 8/07	zpool history
5	Solaris 10 10/08	`gzip`-Komprimierungsalgorithmus
6	Solaris 10 10/08	Pooleigenschaft `bootfs`
7	Solaris 10 10/08	Separate Intent-Log-Geräte
8	Solaris 10 10/08	Delegated Administration
9	Solaris 10 10/08	Eigenschaften `refquota` und `refreservation`
10	Solaris 10 5/09	Cachegeräte
11	Solaris 10 10/09	Verbesserte Bereinigung
12	Solaris 10 10/09	Schnappschusseigenschaften
13	Solaris 10 10/09	Eigenschaft `snapused`
14	Solaris 10 10/09	Eigenschaft `aclinherit` `passthrough-x`
15	Solaris 10 10/09	Speicherplatzberechnung für Benutzer und Gruppen
16	Solaris 10 9/10	Unterstützung der Eigenschaft `stmf`
17	Solaris 10 9/10	RAID-Z-Konfiguration mit dreifacher Parität
18	Solaris 10 9/10	Aufbewahrung von Schnappschüssen
19	Solaris 10 9/10	Entfernen von Loggeräten
20	Solaris 10 9/10	`Komprimierung` `mit` `ZLIe` (Zero-Length Encoding)
21	Solaris 10 9/10	Reserviert
22	Solaris 10 9/10	Empfangene Eigenschaften
23	Solaris 10 8/11	Kleines ZIL
24	Solaris 10 8/11	Systemattribute
25	Solaris 10 8/11	Verbesserte Bereinigungsstatistiken
26	Solaris 10 8/11	Verbesserte Leistung bei der Löschung von Schnappschüssen
27	Solaris 10 8/11	Verbesserte Leistung bei der Erstellung von Schnappschüssen
28	Solaris 10 8/11	Ersetzungen mehrerer virtueller Geräte (vdev)

Version	Solaris 10	Beschreibung
29	Solaris 10 8/11	RAID-Z/Spiegelungs-Hybrid-Zuordner
30	Solaris 10 1/13	Reserviert
31	Solaris 10 1/13	Verbesserte `zfs list`-Leistung
32	Solaris 10 1/13	Ein-MB-Blockgröße

ZFS-Dateisystemversionen

In der folgenden Tabelle sind die ZFS-Dateisystemversionen aufgelistet, die in der Oracle Solaris-Version zur Verfügung stehen. Beachten Sie, dass eine Funktion, die in einer bestimmten Dateisystemversion vorhanden ist, eine bestimmte Poolversion erfordert.

Version	Solaris 10	Beschreibung
1	Solaris 10 6/06	Ursprüngliche ZFS-Dateisystemversionen
2	Solaris 10 10/08	Erweiterte Verzeichniseinträge
3	Solaris 10 10/08	Eindeutiger Bezeichner für Unabhängigkeit von Groß-/Kleinschreibung und das Dateisystem (FUID)
4	Solaris 10 10/09	`userquota`- und `groupquota`-Eigenschaften
5	Solaris 10 8/11	Systemattribute

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

T

U

V

W

X

Z