Guide d'administration Oracle® Solaris ZFS

Référence : E38887–03
Juin 2013

ORACLE

130704@25097

Table des matières

Préface 11

1 Système de fichiers Oracle Solaris ZFS (introduction) 15
Nouveautés de ZFS 15
Amélioration d'utilisation des commandes ZFS 16
Améliorations des instantanés ZFS 16
Propriété `aclmode` améliorée 17
Fonctions d'installation d'Oracle Solaris ZFS 17
Améliorations apportées au flux envoyé par ZFS 17
Différences des instantanés ZFS (`zfs diff`) 18
Récupération de pool de stockage ZFS et améliorations apportées aux performances 18
Réglage du comportement synchrone ZFS 19
Messages du pool ZFS améliorés 19
Améliorations de l'interopérabilité ACL ZFS 21
Scission d'un pool de stockage ZFS mis en miroir (`zpool split`) 22
Nouveau processus du système de fichiers ZFS 22
Améliorations du remplacement des périphériques ZFS 22
Prise en charge de l'installation de ZFS et Flash 24
Migration de zone dans un environnement ZFS 24
Prise en charge de l'installation et de l'initialisation de ZFS 24
Gestion Web ZFS 25
Description d'Oracle Solaris ZFS 26
Stockage ZFS mis en pool 26
Sémantique transactionnelle 26
Sommes de contrôle et données d'autorétablissement 27
Evolutitivé inégalée 27
Instantanés ZFS 28
Administration simplifiée 28

Terminologie ZFS 28
Exigences d'attribution de noms de composants ZFS 31
Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques 32
Granularité du système de fichiers ZFS 32
Comptabilisation de l'espace disque ZFS 32
Montage de système de fichiers ZFS 34
Gestion de volumes classique 35
Modèle ACL Solaris basé sur NFSv4 35

2 Mise en route d'Oracle Solaris ZFS 37
Profils de droits ZFS 37
Exigences et recommandations en matière de matériel et de logiciel ZFS 38
Création d'un système de fichiers ZFS de base 38
Création d'un pool de stockage ZFS de base 39
▼ Identification des exigences de stockage du pool de stockage ZFS 39
▼ Création d'un pool de stockage ZFS 40
Création d'une hiérarchie de systèmes de fichiers ZFS 41
▼ Détermination de la hiérarchie du système de fichiers ZFS 41
▼ Création de systèmes de fichiers ZFS 42

3 Gestion des pools de stockage Oracle Solaris ZFS 45
Composants d'un pool de stockage ZFS 45
Utilisation de disques dans un pool de stockage ZFS 45
Utilisation de tranches dans un pool de stockage ZFS 46
Utilisation de fichiers dans un pool de stockage ZFS 48
Remarques relatives aux pools de stockage ZFS 48
Fonctions de réplication d'un pool de stockage ZFS 49
Configuration de pool de stockage mis en miroir 49
Configuration de pool de stockage RAID-Z 49
Pool de stockage ZFS hybride 51
Données d'autorétablissement dans une configuration redondante 51
Entrelacement dynamique dans un pool de stockage 51
Création et destruction de pools de stockage ZFS 52
Création de pools de stockage ZFS 52
Affichage des informations d'un périphérique virtuel de pool de stockage 59

Gestion d'erreurs de création de pools de stockage ZFS 60
Destruction de pools de stockage ZFS 63
Gestion de périphériques dans un pool de stockage ZFS 64
Ajout de périphériques à un pool de stockage 64
Connexion et séparation de périphériques dans un pool de stockage 69
Création d'un pool par scission d'un pool de stockage ZFS mis en miroir 71
Mise en ligne et mise hors ligne de périphériques dans un pool de stockage 74
Effacement des erreurs de périphérique de pool de stockage 77
Remplacement de périphériques dans un pool de stockage 77
Désignation des disques hot spare dans le pool de stockage 80
Gestion des propriétés de pool de stockage ZFS 85
Requête d'état de pool de stockage ZFS 88
Affichage des informations des pools de stockage ZFS 88
Visualisation des statistiques d'E/S des pools de stockage ZFS 92
Détermination de l'état de maintenance des pools de stockage ZFS 94
Migration de pools de stockage ZFS 100
Préparatifs de migration de pool de stockage ZFS 100
Exportation d'un pool de stockage ZFS 101
Définition des pools de stockage disponibles pour importation 101
Importation de pools de stockage ZFS à partir d'autres répertoires 103
Importation de pools de stockage ZFS 103
Récupération de pools de stockage ZFS détruits 107
Mise à niveau de pools de stockage ZFS 109

4 Installation et initialisation d'un système de fichiers root ZFS Oracle Solaris 111
Installation et initialisation d'un système de fichiers root Oracle Solaris ZFS (présentation) .. 112
Fonctions d'installation de ZFS 112
Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS 113
Installation d'un système de fichiers root ZFS (installation initiale d'Oracle Solaris) 116
▼ Création d'un pool root ZFS mis en miroir (post-installation) 122
Installation d'un système de fichiers root ZFS (installation d'archive Flash Oracle Solaris) 124
Installation d'un système de fichiers root ZFS (installation JumpStart) 128
Mots-clés JumpStart pour ZFS 128
Exemples de profils JumpStart pour ZFS 130

Problèmes JumpStart pour ZFS 131
Migration d'un système de fichiers root ZFS ou mise à jour d'un système de fichiers root ZFS (Live Upgrade) 132
Problèmes de migration ZFS avec Live Upgrade 133
Utilisation de Live Upgrade pour migrer ou mettre à jour un système de fichiers root ZFS (sans zones) 134
Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08) 142
Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure) 147
Gestion de vos périphériques de swap et de vidage ZFS 158
Ajustement de la taille de vos périphériques de swap et de vidage ZFS 159
Personnalisation des volumes de swap et de vidage ZFS 161
Dépannage du périphérique de vidage ZFS 161
Initialisation à partir d'un système de fichiers root ZFS 162
Initialisation à partir d'un disque alternatif d'un pool root ZFS mis en miroir 163
SPARC : initialisation à partir d'un système de fichiers root ZFS 164
x86 : initialisation à partir d'un système de fichiers root ZFS 166
Résolution de problèmes de point de montage empêchant l'initialisation (Solaris 10 10/08) 167
Initialisation à des fins de récupération dans un environnement root ZFS 168
Restauration du pool root ZFS ou des instantanés du pool root 170
▼ Remplacement d'un disque dans le pool root ZFS 170
▼ Création d'instantanés de pool root 172
▼ Recréation d'un pool root ZFS et restauration d'instantanés de pool root 174
▼ Restauration des instantanés d'un pool root à partir d'une initialisation de secours 176

5 Gestion des systèmes de fichiers Oracle Solaris ZFS 179
Gestion des systèmes de fichiers ZFS (présentation) 179
Création, destruction et renommage de systèmes de fichiers ZFS 180
Création d'un système de fichiers ZFS 180
Destruction d'un système de fichiers ZFS 181
Modification du nom d'un système de fichiers ZFS 182
Présentation des propriétés ZFS 183
Propriétés ZFS natives en lecture seule 192
Propriétés ZFS natives définies 193

Propriétés ZFS définies par l'utilisateur 196
Envoi de requêtes sur les informations des systèmes de fichiers ZFS 198
Affichage des informations de base des systèmes ZFS 198
Création de requêtes ZFS complexes 199
Gestion des propriétés ZFS 200
Définition des propriétés ZFS 200
Héritage des propriétés ZFS 201
Envoi de requêtes sur les propriétés ZFS 202
Montage de système de fichiers ZFS 205
Gestion des points de montage ZFS 206
Montage de système de fichiers ZFS 208
Utilisation de propriétés de montage temporaires 209
Démontage des systèmes de fichiers ZFS 209
Activation et annulation du partage des systèmes de fichiers ZFS 210
Définition des quotas et réservations ZFS 212
Définitions de quotas sur les systèmes de fichiers ZFS 213
Définition de réservations sur les systèmes de fichiers ZFS 216
Mise à niveau des systèmes de fichiers ZFS 218

6 Utilisation des instantanés et des clones ZFS Oracle Solaris 219
Présentation des instantanés ZFS 219
Création et destruction d'instantanés ZFS 220
Affichage et accès des instantanés ZFS 223
Restauration d'un instantané ZFS 225
Identification des différences entre des instantanés ZFS (`zfs diff`) 225
Présentation des clones ZFS 227
Création d'un clone ZFS 227
Destruction d'un clone ZFS 228
Remplacement d'un système de fichiers ZFS par un clone ZFS 228
Envoi et réception de données ZFS 229
Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde 230
Reconnaissance des flux d'instantané ZFS 230
Envoi d'un instantané ZFS 232
Réception d'un instantané ZFS 233
Application de différentes valeurs de propriété à un flux d'instantané ZFS 235

Envoi et réception de flux d'instantanés ZFS complexes 237
Réplication distante de données ZFS 239

7 Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS 241
Modèle ACL Solaris 241
Descriptions de syntaxe pour la configuration des ACL 243
Héritage d'ACL 246
Propriétés ACL 247
Configuration d'ACL dans des fichiers ZFS 248
Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé 251
Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé 255
Configuration et affichage d'ACL dans des fichiers ZFS en format compact 261

8 Administration déléguée de ZFS dans Oracle Solaris 267
Présentation de l'administration déléguée de ZFS 267
Désactivation des droits délégués de ZFS 268
Délégation d'autorisations ZFS 268
Délégation des autorisations ZFS (`zfs allow`) 271
Suppression des autorisations déléguées de ZFS (`zfs unallow`) 272
Délégation d'autorisations ZFS (exemples) 272
Affichage des autorisations ZFS déléguées (exemples) 276
Suppression des autorisations ZFS déléguées (exemples) 278

9 Rubriques avancées Oracle Solaris ZFS 281
Volumes ZFS 281
Utilisation d'un volume ZFS en tant que périphérique de swap ou de vidage 282
Utilisation d'un volume ZFS en tant que cible iSCSI Solaris 283
Utilisation de ZFS dans un système Solaris avec zones installées 284
Ajout de systèmes de fichiers ZFS à une zone non globale 285
Délégation de jeux de données à une zone non globale 286
Ajout de volumes ZFS à une zone non globale 287
Utilisation de pools de stockage ZFS au sein d'une zone 288
Gestion de propriétés ZFS au sein d'une zone 288
Explication de la propriété `zoned` 289

Utilisation de pools root ZFS de remplacement 290
Création de pools root de remplacement ZFS 291
Importation de pools root de remplacement 291

10 Dépannage d'Oracle Solaris ZFS et récupération de pool 293
Identification des problèmes ZFS 293
Résolution des problèmes matériels généraux 294
Identification des pannes du matériel et des périphériques 294
Génération de rapports système de messages d'erreur ZFS 296
Identification des problèmes avec les pools de stockage ZFS 296
Recherche de problèmes éventuels dans un pool de stockage ZFS 298
Consultation de la sortie de `zpool status` 298
Résolution des problèmes des périphériques de stockage ZFS 301
Résolution d'un périphérique manquant ou supprimé 301
Remplacement ou réparation d'un périphérique endommagé 304
Résolution des problèmes de système de fichiers ZFS 315
Résolution des problèmes de données dans un pool de stockage ZFS 315
Contrôle de l'intégrité d'un système de fichiers ZFS 315
Données ZFS altérées 318
Résolution des problèmes d'espace ZFS 318
Réparation de données endommagées 320
Réparation d'une configuration ZFS endommagée 325
Réparation d'un système impossible à réinitialiser 325

11 Pratiques recommandées pour Oracle Solaris ZFS 327
Pratiques recommandées pour les pools de stockage 327
Pratiques recommandées générales 327
Pratiques de création de pools de stockage ZFS 329
Pratiques recommandées pour l'optimisation des performances des pools de stockage .. 333
Pratiques recommandées pour la maintenance et la gestion d'un pool de stockage ZFS .. 333
Pratiques recommandées pour les systèmes de fichiers 335
Pratiques recommandées pour la création de systèmes de fichiers 335
Pratiques recommandées pour la surveillance de systèmes de fichiers ZFS 336

A Descriptions des versions d'Oracle Solaris ZFS 337
Présentation des versions ZFS 337
Versions de pool ZFS 337
Versions du système de fichiers ZFS 339

Index 341

Préface

Le *Guide d'administration Oracle Solaris ZFS* fournit des informations sur la configuration et la gestion des systèmes de fichiers ZFS Oracle Solaris.

Ce guide contient des informations sur les systèmes SPARC et x86.

Remarque – Cette version d'Oracle Solaris prend en charge les systèmes utilisant les architectures de processeur SPARC et x86. Les systèmes pris en charge sont répertoriés à la page *Oracle Solaris Hardware Compatibility List* disponible à l'adresse http://www.oracle.com/webfolder/technetwork/hcl/index.html. Ce document présente les différences d'implémentation en fonction des divers types de plates-formes.

Utilisateurs de ce manuel

Ce guide est destiné à toute personne souhaitant configurer et gérer des systèmes de fichiers ZFS Oracle Solaris. Il est recommandé de savoir utiliser le système d'exploitation Oracle Solaris ou toute autre version UNIX.

Organisation de ce document

Le tableau suivant décrit les chapitres de ce document.

Chapitre	Description
Chapitre 1, “Système de fichiers Oracle Solaris ZFS (introduction)”	Présente ZFS, ses fonctionnalités et ses avantages. Il aborde également des concepts de base, ainsi que la terminologie.
Chapitre 2, “Mise en route d'Oracle Solaris ZFS”	Décrit étape par étape les instructions d'une configuration ZFS de base contenant des pools et des systèmes de fichiers simples. Ce chapitre indique également le matériel et logiciels requis pour la création de systèmes de fichiers ZFS.
Chapitre 3, “Gestion des pools de stockage Oracle Solaris ZFS”	Décrit en détail les méthodes de création et d'administration de pools de stockage ZFS.

Chapitre	Description
Chapitre 4, “Installation et initialisation d'un système de fichiers root ZFS Oracle Solaris”	Décrit la procédure d'installation et d'initialisation d'un système de fichiers ZFS. La migration d'un système de fichiers root UFS vers un système de fichiers root ZFS à l'aide de Solaris Live Upgrade est également abordée.
Chapitre 5, “Gestion des systèmes de fichiers Oracle Solaris ZFS”	Décrit en détail les méthodes de gestion de systèmes de fichiers ZFS. Ce chapitre décrit des concepts tels que la disposition hiérarchique de systèmes de fichiers, l'héritage de propriétés, la gestion automatique de points de montage et les interactions de partage.
Chapitre 6, “Utilisation des instantanés et des clones ZFS Oracle Solaris”	Décrit les méthodes de création et d'administration d'instantanés ZFS et de clones.
Chapitre 7, “Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS”	Explique comment utiliser des listes de contrôle d'accès (ACL, Access Control List) pour la protection des fichiers ZFS en fournissant des autorisations à un niveau de granularité plus fin que les autorisations UNIX standard.
Chapitre 8, “Administration déléguée de ZFS dans Oracle Solaris”	Explique comment utiliser les fonctions de l'administration déléguée de ZFS pour permettre aux utilisateurs ne disposant pas des autorisations nécessaires d'effectuer des tâches d'administration ZFS.
Chapitre 9, “Rubriques avancées Oracle Solaris ZFS”	Explique comment utiliser des volumes et des systèmes de fichiers ZFS dans un système Oracle Solaris comportant des zones et comment utiliser les pools root de remplacement.
Chapitre 10, “Dépannage d'Oracle Solaris ZFS et récupération de pool”	Explique comment identifier des modes de défaillance de ZFS et les solutions existantes. Les étapes de prévention de ces défaillances sont également abordées.
Chapitre 11, “Pratiques recommandées pour Oracle Solaris ZFS”	Décrit les pratiques recommandées pour la création, la surveillance et la mise à jour de pools de stockage ZFS et de systèmes de fichiers.
Annexe A, “Descriptions des versions d'Oracle Solaris ZFS”	Décrit les versions ZFS disponibles, les fonctionnalités de chacune d'entre elles et le SE Solaris fournissant la version et les fonctionnalités ZFS.

Documentation connexe

Pour obtenir des informations générales sur l'administration de systèmes Oracle Solaris, reportez-vous aux documents suivants :

- *Guide d’administration système : Administration avancée*
- *System Administration Guide: Devices and File Systems*
- *System Administration Guide: Security Services*

Accès aux services de support Oracle

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Conventions typographiques

Le tableau ci-dessous décrit les conventions typographiques utilisées dans ce manuel.

TABLEAU P–1 Conventions typographiques

Type de caractères	Description	Exemple
`AaBbCc123`	Noms des commandes, fichiers et répertoires, ainsi que messages système.	Modifiez votre fichier `.login`. Utilisez `ls -a` pour afficher la liste de tous les fichiers. `nom_machine% Vous avez reçu du courrier.`
`AaBbCc123`	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	`nom_machine%` **`su`** `Mot de passe :`
aabbcc123	Paramètre fictif : à remplacer par un nom ou une valeur réel(le).	La commande permettant de supprimer un fichier est `rm` *filename*.
AaBbCc123	Titres de manuel, nouveaux termes et termes importants.	Reportez-vous au chapitre 6 du *Guide de l'utilisateur*. Un *cache* est une copie des éléments stockés localement. *N'enregistrez pas* le fichier. **Remarque :** en ligne, certains éléments mis en valeur s'affichent en gras.

Invites de shell dans les exemples de commandes

Le tableau suivant présente l'invite système UNIX par défaut et l'invite superutilisateur pour les shells faisant partie du SE Oracle Solaris. Dans les exemples de commandes, l'invite de shell indique si la commande doit être exécutée par un utilisateur standard ou un utilisateur doté des privilèges nécessaires.

TABLEAU P–2 Invites de shell

Shell	Invite
Shell Bash, shell Korn et shell Bourne	`$`
Shell Bash, shell Korn et shell Bourne pour superutilisateur	`#`
C shell	`nom_machine%`
C shell pour superutilisateur	`nom_machine#`

◆ ◆ ◆ CHAPITRE 1

Système de fichiers Oracle Solaris ZFS (introduction)

Ce chapitre présente le système de fichiers ZFS Oracle Solaris, ses fonctions et ses avantages. Il aborde également la terminologie de base utilisée dans le reste de ce document.

Ce chapitre contient les sections suivantes :

- "Nouveautés de ZFS" à la page 15
- "Description d'Oracle Solaris ZFS" à la page 26
- "Terminologie ZFS" à la page 28
- "Exigences d'attribution de noms de composants ZFS" à la page 31
- "Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques" à la page 32

Nouveautés de ZFS

Cette section décrit les nouvelles fonctions du système de fichiers ZFS.

- "Amélioration d'utilisation des commandes ZFS" à la page 16
- "Améliorations des instantanés ZFS" à la page 16
- "Propriété `aclmode` améliorée" à la page 17
- "Fonctions d'installation d'Oracle Solaris ZFS" à la page 17
- "Améliorations apportées au flux envoyé par ZFS" à la page 17
- "Différences des instantanés ZFS (`zfs diff`)" à la page 18
- "Récupération de pool de stockage ZFS et améliorations apportées aux performances" à la page 18
- "Réglage du comportement synchrone ZFS" à la page 19
- "Messages du pool ZFS améliorés" à la page 19
- "Améliorations de l'interopérabilité ACL ZFS" à la page 21
- "Scission d'un pool de stockage ZFS mis en miroir (`zpool split`) " à la page 22
- "Nouveau processus du système de fichiers ZFS" à la page 22
- "Prise en charge de l'installation de ZFS et Flash" à la page 24
- "Gestion Web ZFS" à la page 25

Amélioration d'utilisation des commandes ZFS

Oracle Solaris 10 1/13 : les commandes `zfs` et `zpool` disposent d'une sous-commande `help` qui peut fournir des informations sur les sous-commandes `zfs` et `zpool`, ainsi que sur leurs options respectives. Par exemple :

```
# 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>
```

Pour plus d'informations, reportez-vous aux pages de manuel `zfs`(1M) et `zpool`(1M).

Améliorations des instantanés ZFS

Oracle Solaris 10 1/13 : cette version inclut les améliorations d'instantanés ZFS suivantes :

- La commande `zfs snapshot` dispose d'un alias `snap` qui offre une syntaxe abrégée pour cette commande. Par exemple :

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

- La commande `zfs diff` offre une option d'énumération, `-e`, pour identifier tous les fichiers ajoutés ou modifiés entre les deux instantanés. La sortie générée identifie tous les fichiers ajoutés, mais ne fournit pas les suppressions possibles. Par exemple :

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

  Vous pouvez aussi utiliser l'option `-o` pour identifier les champs sélectionnés à afficher. Par exemple :

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

Pour plus d'informations sur la création d'instantanés ZFS, reportez-vous au Chapitre 6, “Utilisation des instantanés et des clones ZFS Oracle Solaris”.

Propriété `aclmode` améliorée

Oracle Solaris 10 1/13 : la propriété `aclmode` modifie le comportement des ACL quand des autorisations ACL sur un fichier sont modifiées pendant une opération `chmod`. La propriété `aclmode` a été réintroduite avec les valeurs suivantes :

- `discard` :un système de fichiers dont la valeur de la propriété `aclmode` est `discard` supprime toutes les entrées d'ACL qui ne représentent pas le mode du fichier. Il s'agit de la valeur par défaut.
- `mask` : un système de fichiers dont la valeur de la propriété `aclmode` est `mask` restreint les autorisations utilisateur ou groupe. Les autorisations sont réduites de manière à ne pas excéder les bits d'autorisation du groupe, à moins qu'il ne s'agisse d'une entrée utilisateur possédant le même UID que le propriétaire du fichier ou du répertoire. Dans ce cas, les autorisations d'ACL sont réduites de manière à ne pas excéder les bits d'autorisation du propriétaire. La valeur de masque préserve en outre l'ACL lors des modifications de mode successives, à condition qu'aucune opération de jeu d'ACL explicite n'ait été effectuée.
- `passthrough` : un système de fichiers avec une propriété `aclmode` de `passthrough` indique qu'aucune modification n'est apportée à l'ACL en dehors de la génération des entrées d'ACL nécessaires pour représenter le nouveau mode du fichier ou du répertoire.

Pour plus d'informations, reportez-vous à l'Exemple 7–13.

Fonctions d'installation d'Oracle Solaris ZFS

Oracle Solaris 10 8/11 : dans cette version, les nouvelles fonctions d'installation suivantes sont disponibles :

- Vous pouvez utiliser la méthode d'installation en mode Texte pour installer un système avec une archive Flash ZFS. Pour plus d'informations, reportez-vous à l'Exemple 4–3.
- Vous pouvez utiliser la commande Oracle Solaris Live Upgrade `luupgrade` pour installer une archive Flash root ZFS. Pour plus d'informations, reportez-vous à l'Exemple 4–8.
- Vous pouvez utiliser la commande `lucreate` Oracle Solaris Live Upgrade pour spécifier un système de fichiers /var distinct. Pour plus d'informations, reportez-vous à l'Exemple 4–5.

Améliorations apportées au flux envoyé par ZFS

Oracle Solaris 10 8/11 : dans cette version, vous pouvez définir les propriétés du système de fichiers qui sont envoyées et reçues dans un flux d'instantané. Ces améliorations offrent

davantage de flexibilité pour appliquer des propriétés du système de fichiers dans un flux envoyé à un système de fichiers récepteur ou pour déterminer si les propriétés du système de fichiers local, telles que la valeur de propriété `mountpoint`, doivent être ignorées lorsqu'elles sont reçues.

Pour plus d'informations, reportez-vous à la section "Application de différentes valeurs de propriété à un flux d'instantané ZFS" à la page 235.

Différences des instantanés ZFS (`zfs diff`)

Oracle Solaris 10 8/11 : dans cette version, vous pouvez déterminer les différences des instantanés ZFS à l'aide de la commande `zfs diff`.

Supposons par exemple que les deux instantanés suivants sont créés :

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

Par exemple, afin d'identifier les différences entre deux instantanés, utilisez une syntaxe semblable à la suivante :

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

Dans la sortie, `M` indique que le répertoire a été modifié. Le `+` indique que `fileB` existe dans l'instantané le plus récent.

Pour plus d'informations, reportez-vous à la section "Identification des différences entre des instantanés ZFS (`zfs diff`)" à la page 225.

Récupération de pool de stockage ZFS et améliorations apportées aux performances

Oracle Solaris 10 8/11 : dans cette version, les nouvelles fonctions du pool de stockage ZFS suivantes sont fournies :

- Vous pouvez importer un pool avec un journal manquant en utilisant la commande `zpool import -m`. Pour plus d'informations, reportez-vous à la section "Importation d'un pool avec un périphérique de journalisation manquant" à la page 105.

- Vous pouvez importer un pool en mode lecture seule. Cette fonction est principalement destinée à la récupération de pool. Si un pool endommagé n'est pas accessible car les périphériques sous-jacents le sont également, vous pouvez importer le pool en lecture seule pour récupérer les données. Pour plus d'informations, reportez-vous à la section "Importation d'un pool en mode lecture seule" à la page 106.
- Un pool de stockage RAID-Z (`raidz1`, `raidz2` ou `raidz3`) créé dans cette version comportera des métadonnées sensibles à la latence qui seront automatiquement mises en miroir pour améliorer les performances de capacité de traitement d'E/S de lecture. Pour les pools RAID-Z existants mis à niveau vers la version 29 ou une version ultérieure du pool, un certain nombre de métadonnées seront mises en miroir pour toutes les données nouvellement écrites.

  Les métadonnées mises en miroir dans un pool RAID-Z ne fournissent *pas* de protection supplémentaire contre les pannes matérielles, comme c'est le cas pour un pool de stockage mis en miroir. Les métadonnées mises en miroir consomment de l'espace supplémentaire, mais la protection RAID-Z est la même que dans les versions précédentes. Cette amélioration est destinée à l'amélioration des performances uniquement.

Réglage du comportement synchrone ZFS

Solaris 10 8/11 : dans cette version, vous pouvez déterminer un comportement synchrone du système de fichiers ZFS à l'aide de la propriété `sync`.

Le comportement synchrone par défaut consiste à écrire toutes les transactions des systèmes de fichiers synchrones dans le journal de tentatives et à vider tous les périphériques pour s'assurer que les données sont stables. La désactivation du comportement synchrone par défaut n'est pas recommandée. Elle pourrait avoir des répercussions sur les applications qui dépendent de la prise en charge synchrone et risquerait d'entraîner des pertes de données.

La propriété `sync` peut être définie avant ou après la création du système de fichiers. Dans tous les cas, la valeur de propriété prend effet immédiatement. Par exemple :

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

Le paramètre `zil_disable` n'est plus disponible dans les versions Oracle Solaris incluant la propriété `sync`.

Pour plus d'informations, reportez-vous au Tableau 5–1.

Messages du pool ZFS améliorés

Oracle Solaris10 8/11 : dans cette version, vous pouvez utiliser l'option `-T` afin de fournir un intervalle et une valeur de comptage pour les commandes `zpool list` et `zpool status` pour l'affichage d'informations supplémentaires.

En outre, des nettoyages du pool et des informations de réargenture supplémentaires sont disponibles via la commande `zpool status` comme suit :

- Rapport de progression de la réargenture. Par exemple :

```
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
```

- Rapport de progression du nettoyage. Par exemple :

```
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
```

- Message de fin de la réargenture. Par exemple :

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

- Message de fin du nettoyage. Par exemple :

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

- Message d'annulation du nettoyage en cours. Par exemple :

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

- Les messages de fin de la réargenture et du nettoyage subsistent après plusieurs réinitialisation du système.

La syntaxe suivante utilise l'intervalle et l'option de comptage pour afficher en permanence les informations relatives à la réargenture du pool en cours. Vous pouvez utiliser la valeur `-T d` pour afficher les informations au format de date standard ou `-T u` pour les afficher dans un format interne.

```
# 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
```

Améliorations de l'interopérabilité ACL ZFS

Oracle Solaris 10 8/11 : dans cette version, les améliorations ACL suivantes sont fournies :

- Les ACL triviales ne requièrent pas d'entrée de contrôle d'accès (ACE) `de refus`, à l'exception des autorisations extraordinaires. Par exemple, un mode `0644`, `0755` ou `0664` ne requiert pas d'ACE `de refus`, tandis qu'un mode `0705`, `0060`, et ainsi de suite, requiert des ACE `de refus`.

  L'ancien comportement inclut des ACE `de refus` dans une ACL triviale telle que `644`. Par exemple :

```
# 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
```

  Le nouveau comportement pour une ACL triviale telle que `644` n'inclut pas les ACE `de refus`. Par exemple :

```
# 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
```

- Les ACL ne plus sont scindées en plusieurs ACE pendant l'héritage pour tenter de conserver l'autorisation d'origine inchangée. Au lieu de cela, les autorisations sont modifiées de manière à appliquer le mode de création de fichier.
- Le comportement de la propriété `aclinherit` inclut une réduction des autorisations lorsque la propriété est définie sur `restricted` (restreint), ce qui signifie que les ACL ne sont plus scindées en plusieurs ACE pendant l'héritage.
- Une nouvelle règle de calcul du mode d'autorisation signifie que si une ACL possède une entrée de contrôle d'accès *user* qui est également propriétaire du fichier, alors ces autorisations sont incluses dans le calcul du mode d'autorisation. La même règle s'applique lorsqu'une entrée de contrôle d'accès *group* est propriétaire de groupe du fichier.

Pour plus d'informations, reportez-vous au Chapitre 7, “Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS”.

Scission d'un pool de stockage ZFS mis en miroir (`zpool split`)

Oracle 10 9/10 : dans cette version de Solaris, vous pouvez utiliser la commande `zpool split` pour scinder un pool de stockage mis en miroir, ce qui déconnecte un ou plusieurs disques dans le pool mis en miroir d'origine pour créer un autre pool identique.

Pour plus d'informations, reportez-vous à la section "Création d'un pool par scission d'un pool de stockage ZFS mis en miroir" à la page 71.

Nouveau processus du système de fichiers ZFS

Oracle Solaris 10 9/10 : dans cette version, chaque pool de stockage ZFS est associé à un processus, `zpool-`*poolname*. Les threads dans ce processus sont les threads de traitement d'E/S du pool permettant de gérer les tâches d'E/S, telles la compression et la validation de la somme de contrôle, associées au pool. Le but de ce processus est d'indiquer l'utilisation de la CPU de chaque pool de stockage.

Des informations relatives à ces processus en cours d'exécution peuvent être consultées à l'aide des commandes `ps` et `prstat`. Ces processus sont uniquement disponibles dans la zone globale. Pour de plus amples informations, reportez-vous à la section `SDC(7)`.

Améliorations du remplacement des périphériques ZFS

Oracle Solaris10 9/10 : dans cette version, un événement du système ou *sysevent* est fourni lorsque les disques dans un pool sont remplacées par des disques plus volumineux. ZFS a été amélioré pour reconnaître ces événements et ajuste le pool en fonction de la nouvelle taille du disque, selon la définition de la propriété `autoexpand`. Vous pouvez utiliser la propriété de pool `autoexpand` pour activer ou désactiver l'expansion automatique du pool lorsqu'un disque plus important remplace un disque moins volumineux.

Ces améliorations vous permettent d'augmenter la taille du pool sans avoir à exporter et importer un pool, ni à réinitialiser le système.

Par exemple, l'extension automatique du LUN est activée sur le pool `tank`.

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

Vous pouvez également créer le pool sans activer la propriété `autoexpand`.

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

La propriété `autoexpand` est désactivée par défaut, de sorte que vous pouvez décider si vous souhaitez que la taille du pool soit agrandie lorsqu'un disque plus important remplace un disque moins volumineux.

Vous pouvez également agrandir la taille d'un pool à l'aide de la commande `zpool online -e`. Par exemple :

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

Vous pouvez également réinitialiser la propriété `autoexpand` après avoir connecté ou mis à disposition un disque plus important en utilisant la commande `zpool replace`. Par exemple, le pool suivant est créé avec un disque de 8 Go (`c0t0d0`). Le disque de 8 Go est remplacé par un disque de 16 Go (`c1t13d0`). Toutefois, la taille du pool ne sera pas étendue jusqu'à l'activation de la propriété `autoexpand`.

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

Vous pouvez également agrandir le disque sans activer la propriété `autoexpand` en utilisant la commande `zpool online -e` même si le périphérique est déjà en ligne. Par exemple :

```
# 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  -
```

Cette version propose les améliorations supplémentaires de remplacement de périphériques suivantes :

- Dans les versions précédentes, le système de fichiers ZFS ne pouvait pas remplacer un disque existant par un autre disque ni connecter un disque si la taille du disque de remplacement était légèrement différente. Dans cette version, vous pouvez remplacer un disque existant par un autre disque ou connecter un nouveau disque d'à peu près la même taille, à condition que le pool ne soit pas déjà plein.

- Dans cette version, vous ne devez pas réinitialiser le système ou exporter/importer un pool pour étendre la taille du pool. Comme décrit ci-dessus, vous pouvez activer la propriété `autoexpand` ou utiliser la commande `zpool online -e` pour étendre le disque.

Pour obtenir des informations sur le remplacement de périphériques, reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 77.

Prise en charge de l'installation de ZFS et Flash

Solaris 10 10/09 : dans cette version, vous pouvez définir un profil JumpStart pour identifier une archive Flash de pool root ZFS. Pour plus d'informations, reportez-vous à la section “Installation d'un système de fichiers root ZFS (installation d'archive Flash Oracle Solaris)” à la page 124.

Migration de zone dans un environnement ZFS

Solaris 10 5/09 : cette version étend la prise en charge des zones de migration dans un environnement ZFS avec Oracle Solaris Live Upgrade. Pour plus d'informations, reportez-vous à la section “Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)” à la page 147.

Vous trouverez la liste des problèmes connus relatifs à cette version dans les notes de version de Solaris 10 5/09.

Prise en charge de l'installation et de l'initialisation de ZFS

Solaris 10 10/08 : cette version permet d'installer et d'initialiser un système de fichiers root ZFS. Vous pouvez utiliser l'option d'installation initiale ou la fonction JumpStart pour installer un système de fichiers root ZFS. Vous pouvez également utiliser la fonction Oracle Solaris Live Upgrade pour migrer d'un système de fichiers root UFS vers un système de fichiers root ZFS. La prise en charge ZFS des périphériques de swap et de vidage est également disponible. Pour plus d'informations, reportez-vous au Chapitre 4, “Installation et initialisation d'un système de fichiers root ZFS Oracle Solaris”.

Vous trouverez la liste des problèmes connus relatifs à cette version dans les notes de version de Solaris 10 10/08.

Gestion Web ZFS

Solaris 10 6/06 : la console d'administration ZFS est un outil de gestion ZFS basé sur le Web permettant d'effectuer les tâches administratives suivantes :

- Créer un nouveau pool de stockage.
- Augmenter la capacité d'un pool existant.
- Déplacer (exporter) un pool de stockage vers un autre système.
- Importer un pool de stockage précédemment exporté afin qu'il soit disponible dans un autre système.
- Visualiser des informations sur les pools de stockage.
- Créer un système de fichiers PCFS.
- Créer un volume.
- Créer un instantané d'un système de fichiers ou d'un volume.
- Restaurer un système de fichiers à l'aide d'un instantané précédent.

Vous pouvez accéder à la console d'administration ZFS par le biais d'un navigateur Web sécurisé en vous connectant à l'adresse URL suivante :

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

Si vous avez saisi l'adresse URL correcte mais ne parvenez pas à atteindre la console d'administration ZFS, il est possible que le serveur ne soit pas démarré. Pour démarrer le serveur, exécutez la commande suivante :

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

Pour exécutez le serveur automatiquement à l'initialisation du système, tapez la commande suivante :

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

Remarque – Les systèmes de fichiers et les pools de stockage ZFS ne peuvent pas être gérés à l'aide de Solaris Management Console (`smc`).

Description d'Oracle Solaris ZFS

Le système de fichiers ZFS présente des fonctions et des avantages uniques au monde. Ce système de fichiers modifie radicalement les méthodes d'administration des systèmes de fichiers. ZFS est robuste, évolutif et facile à administrer.

Stockage ZFS mis en pool

ZFS utilise le concept de *pools de stockage* pour la gestion du stockage physique. Auparavant, l'élaboration des systèmes de fichiers reposait sur un périphérique physique unique. Afin de traiter plusieurs périphériques et d'assurer la redondance de données, le concept de *gestionnaire de volume* a été introduit pour fournir la représentation d'un périphérique. Ainsi, il n'est plus nécessaire de modifier les systèmes de fichiers pour bénéficier de plusieurs périphériques. Cette conception ajoutait un niveau de complexité supplémentaire et empêchait finalement les avancées de certains systèmes de fichiers, car le système de fichiers ne pouvait pas contrôler le placement physique des données dans les volumes virtualisés.

Le système de fichiers ZFS élimine la gestion du volume. Plutôt que de vous obliger à créer des volumes virtualisés, ZFS regroupe les périphériques dans un pool de stockage. Le pool de stockage décrit les caractéristiques physiques du stockage (disposition de périphérique, redondance de données, etc.) et agit en tant qu'espace de stockage de données arbitraires à partir duquel il est possible de créer des systèmes de fichiers. Désormais, les systèmes de fichiers ne sont plus limités à des périphériques individuels. Ainsi, ils peuvent partager l'espace disque avec l'ensemble des systèmes de fichiers du pool. Il n'est plus nécessaire de prédéterminer la taille des systèmes de fichiers, car celle-ci augmente automatiquement au sein de l'espace disque alloué au pool de stockage. En cas d'ajout d'espace de stockage, tous les systèmes de fichiers du pool peuvent immédiatement utiliser l'espace disque supplémentaire, sans requérir des tâches supplémentaires. Le pool de stockage fonctionne de la même manière qu'un système de mémoire virtuelle sous plusieurs aspects : lors de l'ajout d'un module DIMM à un système, le système d'exploitation ne force pas l'exécution de commandes pour configurer la mémoire et pour l'assigner aux processus. Tous les processus du système utilisent automatiquement la mémoire supplémentaire.

Sémantique transactionnelle

ZFS étant un système de fichiers transactionnel, l'état du système de fichiers reste toujours cohérent sur le disque. Les systèmes de fichiers classiques écrasent les données en place. Ainsi, en cas de réduction de la puissance du système, par exemple, entre le moment où un bloc de données est alloué et celui où il est lié à un répertoire, le système de fichiers reste incohérent. Auparavant, la commande `fsck` permettait de résoudre ce problème. Cette commande permettait de vérifier l'état du système de fichiers et de tenter de réparer les incohérences détectées au cours du processus. Les incohérences dans les systèmes de fichiers pouvaient poser

de sérieux problèmes aux administrateurs. La commande `fsck` ne garantissait pas la résolution de tous les problèmes. Plus récemment, les systèmes de fichiers ont introduit le concept de *journalisation*. Le processus de journalisation enregistre les actions dans un journal séparé, lequel peut ensuite être *lu* en toute sécurité en cas de panne du système. Ce processus requiert un temps système inutile car les données doivent être écrites deux fois. En outre, il entraîne souvent d'autres problèmes, par exemple l'impossibilité de relire correctement le journal.

Avec un système de fichiers transactionnel, la gestion de données s'effectue avec une sémantique de *copie lors de l'écriture.* Les données ne sont jamais écrasées et toute séquence d'opération est entièrement validée ou entièrement ignorée. L'altération du système de fichiers en raison d'une coupure de courant ou d'un arrêt du système est impossible. Même s'il se peut que les éléments les plus récents écrits sur les données soient perdus, le système de fichiers reste cohérent. De plus, les données synchrones (écrites avec l'indicateur `O_DSYNC`) sont toujours écrites avant le renvoi. Ainsi, toute perte est impossible.

Sommes de contrôle et données d'autorétablissement

Avec ZFS, toutes les données et métadonnées sont vérifiées selon un algorithme de somme de contrôle sélectionné par l'utilisateur. Les systèmes de fichiers classiques fournissant le contrôle de sommes l'effectuaient par bloc, en raison de la couche de gestion de volumes et de la conception classique de système de fichiers. Le terme classique signifie que certaines pannes, comme l'écriture d'un bloc complet dans un emplacement incorrect, peuvent entraîner des incohérences dans les données, sans pour autant entraîner d'erreur dans les sommes de contrôle. Les sommes de contrôle ZFS sont stockées de façon à détecter ces pannes et à effectuer une récupération de manière appropriée. Toutes les opérations de contrôle de somme et de récupération des données sont effectuées sur la couche du système de fichiers et sont transparentes aux applications.

De plus, ZFS fournit des données d'autorétablissement. ZFS assure la prise en charge de pools de stockage avec différents niveaux de redondance de données. Lorsqu'un bloc de données endommagé est détecté, ZFS récupère les données correctes à partir d'une autre copie redondante et répare les données endommagées en les remplaçant par celles de la copie.

Evolutivité inégalée

L'évolutivité de ZFS représente l'un des éléments clés de sa conception. La taille du système de fichiers lui-même est de 128 bits et vous pouvez utiliser jusqu'à 256 quadrillions de zettaoctets de stockage. L'ensemble des métadonnées est alloué de façon dynamique. Il est donc inutile de pré-allouer des inodes ou de limiter l'évolutivité du système de fichiers lors de sa création. Tous les algorithmes ont été écrits selon cette exigence d'évolutivité. Les répertoires peuvent contenir jusqu'à 2^{48} (256 trillions) d'entrées et le nombre de systèmes de fichiers ou de fichiers contenus dans un système de fichiers est illimité.

Instantanés ZFS

Un *instantané* est une copie en lecture seule d'un système de fichiers ou d'un volume. La création d'instantanés est rapide et facile. Ils n'utilisent initialement aucun espace disque supplémentaire dans le pool.

A mesure que le jeu de données actif est modifié, l'espace disque occupé par l'instantané augmente tandis que l'instantané continue de référencer les anciennes données. Par conséquent, l'instantané évite que les données soit libérées à nouveau dans le pool.

Administration simplifiée

Point le plus important, ZFS fournit un modèle administration qui a été énormément simplifié. Grâce à une disposition hiérarchique des systèmes de fichiers, à l'héritage des propriétés et à la gestion automatique des points de montage et de la sémantique de partage NFS, ZFS facilite la création et la gestion de systèmes de fichiers sans requérir de nombreuses commandes, ni la modification de fichiers de configuration. Vous pouvez définir des quotas ou des réservations, activer ou désactiver la compression ou encore gérer les point de montage pour plusieurs systèmes de fichiers avec une seule commande. Vous pouvez vérifier ou remplacer les périphériques sans devoir apprendre un jeu de commandes de gestion de volumes spécifique. Vous pouvez envoyer et recevoir des flux d'instantanés du système de fichiers.

ZFS assure la gestion des systèmes de fichiers par le biais d'une hiérarchie qui facilite la gestion des propriétés telles que les quotas, les réservations, la compression et les points de montage. Dans ce modèle, les systèmes de fichiers constituent le point de contrôle central. Les systèmes de fichiers eux-mêmes étant très peu coûteux (autant que la création d'un nouveau répertoire), il est recommandé de créer un système de fichiers pour chaque utilisateur, projet, espace de travail, etc. Cette conception permet de définir des points de gestion détaillés.

Terminologie ZFS

Cette section décrit la terminologie de base utilisée dans ce document :

Environnement d'initialisation alternatif	Environnement d'initialisation créé par la commande `lucreate` et éventuellement mis à jour par la commande `luupgrade`. Cependant, il ne s'agit pas de l'environnement d'initialisation principal ou actif. L'environnement d'initialisation secondaire peut devenir l'environnement d'initialisation principal en exécutant la commande `luactivate`.
somme de contrôle (checksum)	Hachage de 256 bits des données dans un bloc de système de données. La fonctionnalité de contrôle

de somme regroupe entre autres, le contrôle de somme simple et rapide fletcher4 (paramètre par défaut), ainsi que les puissantes fonctions de hachage cryptographique telles que SHA256.

clone

Système de fichiers dont le contenu initial est identique à celui d'un instantané.

Pour plus d'informations sur les clones, reportez-vous à la section "Présentation des clones ZFS" à la page 227.

jeu de données

Nom générique pour les composants ZFS suivants : clones, systèmes de fichiers, instantanés et volumes.

Chaque jeu de données est identifié par un nom unique dans l'espace de noms ZFS. Les jeux de données sont identifiés à l'aide du format suivant :

pool/*path*[*@snapshot*]

pool — Identifie le nom d'un pool de stockage contenant le jeu de données.

path — Nom de chemin délimité par slash pour le composant de jeu de données

snapshot — Composant optionnel identifiant l'instantané d'un jeu de données.

Pour plus d'informations sur les jeux de données, reportez-vous au Chapitre 5, "Gestion des systèmes de fichiers Oracle Solaris ZFS".

système de fichiers

Jeu de données ZFS de type `filesystem` monté au sein de l'espace de noms système standard et se comportant comme les autres systèmes de fichiers.

Pour plus d'informations sur les systèmes de fichiers, reportez-vous au Chapitre 5, "Gestion des systèmes de fichiers Oracle Solaris ZFS".

miroir

Périphérique virtuel stockant des copies identiques de données sur un ou plusieurs disques. Lorsqu'un disque d'un miroir est

	défaillant, tout autre disque du miroir est en mesure de fournir les mêmes données.
pool	Groupe logique de périphériques décrivant la disposition et les caractéristiques physiques du stockage disponible. L'espace disque pour les jeux de données est alloué à partir d'un pool. Pour plus d'informations sur les pools de stockage, reportez-vous au Chapitre 3, “Gestion des pools de stockage Oracle Solaris ZFS”.
Environnement d'initialisation principal	Environnement d'initialisation utilisé par la commande `lucreate` pour créer un environnement d'initialisation alternatif. Par défaut, l'environnement d'initialisation principal correspond à l'environnement d'initialisation actuel. Ce paramètre par défaut peut être modifié à l'aide de l'option `lucreate -s`.
RAID-Z	Périphérique virtuel stockant les données et la parité sur plusieurs disques. Pour plus d'informations sur RAID-Z, reportez-vous à la section “Configuration de pool de stockage RAID-Z” à la page 49.
réargenture	Processus de copie de données d'un périphérique à un autre, connu sous le nom de *réargenture*. Par exemple, si un périphérique de miroir est remplacé ou mis hors ligne, les données du périphérique de miroir le plus actuel sont copiées dans le périphérique de miroir nouvellement restauré. Dans les produits de gestion de volumes classiques, ce processus est appelé *réargenture de miroir*. Pour plus d'informations sur la réargenture ZFS, reportez-vous à la section “Affichage de l'état de réargenture” à la page 313.
instantané	Copie ponctuelle en lecture seule d'un système de fichiers ou d'un volume. Pour plus d'informations sur les instantanés, reportez-vous à la section “Présentation des instantanés ZFS” à la page 219.

périphérique virtuel	Périphérique logique dans un pool. il peut s'agir d'un périphérique physique, d'un fichier ou d'une collection de périphériques. Pour plus d'informations sur les périphériques virtuels, reportez-vous à la section "Affichage des informations d'un périphérique virtuel de pool de stockage" à la page 59.
volume	Jeu de données représentant un périphérique en mode bloc. Vous pouvez par exemple créer un volume ZFS en tant que périphérique de swap. Pour plus d'informations sur les volumes ZFS, reportez-vous à la section "Volumes ZFS" à la page 281.

Exigences d'attribution de noms de composants ZFS

L'attribution de noms de chaque composant ZFS, tels que les jeux de données et les pools, doit respecter les règles suivantes :

- Chaque composant ne peut contenir que des caractères alphanumériques en plus des quatre caractères spéciaux suivants :
  - Soulignement (_)
  - Trait d'union (-)
  - Deux points (:)
  - Point (.)
- Les noms de pool doivent commencer par une lettre et peuvent uniquement contenir des caractères alphanumériques, ainsi que des caractères de soulignement (_), des tirets (-) et des points (.). Voici les restrictions concernant les noms de pool :
  - La séquence de début `c[0-9]` n'est pas autorisée.
  - Le nom `log` est réservé.
  - Vous ne pouvez pas utiliser un nom commençant par `mirror`, `raidz` , `raidz1`, `raidz2`, `raidz3` ou `spare` car ces noms sont réservés.
  - Les noms de pools ne doivent pas contenir le signe de pourcentage (`%`).
- Les noms de jeux de données doivent commencer par un caractère alphanumérique.
- Les noms de jeux de données ne doivent pas contenir le signe de pourcentage (`%`).

De plus, les composants vides ne sont pas autorisés.

Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques

- "Granularité du système de fichiers ZFS" à la page 32
- "Comptabilisation de l'espace disque ZFS" à la page 32
- "Montage de système de fichiers ZFS" à la page 34
- "Gestion de volumes classique" à la page 35
- "Modèle ACL Solaris basé sur NFSv4" à la page 35

Granularité du système de fichiers ZFS

Traditionnellement, les systèmes de fichiers étaient restreints à un périphérique et par conséquent à la taille de ce périphérique. Les créations successives de systèmes de fichiers classiques dues aux contraintes de taille demandent du temps et s'avèrent parfois difficile. Les produits de gestion de volume traditionnels aident à gérer ce processus.

Les systèmes de fichiers ZFS n'étant pas limités à des périphériques spécifiques, leur création est facile et rapide, tout comme celle des répertoires. La taille des systèmes de fichiers ZFS augmente automatiquement dans l'espace disque alloué au pool de stockage sur lequel ils se trouvent.

Au lieu de créer un système de fichiers, comme `/export/home`, pour la gestion de plusieurs sous-répertoires d'utilisateurs, vous pouvez créer un système de fichiers par utilisateur. Vous pouvez facilement définir et gérer plusieurs systèmes de fichiers en appliquant des propriétés pouvant être héritées par le système de fichiers descendant au sein de la hiérarchie.

Pour obtenir un exemple de création d'une hiérarchie de système de fichiers, reportez-vous à la section "Création d'une hiérarchie de systèmes de fichiers ZFS" à la page 41.

Comptabilisation de l'espace disque ZFS

Le système de fichiers ZFS repose sur le concept de stockage de pools. Contrairement aux systèmes de fichiers classiques, qui sont mappés vers un stockage physique, tous les systèmes de fichiers ZFS d'un pool partagent le stockage disponible dans le pool. Ainsi, l'espace disponible indiqué par des utilitaires tels que `df` peut changer alors même que le système de fichiers est inactif, parce que d'autres systèmes de fichiers du pool utilisent ou libèrent de l'espace.

Notez que la taille maximale du système de fichiers peut être limitée par l'utilisation des quotas. Pour obtenir des informations sur les quotas, reportez-vous à la section "Définitions de quotas sur les systèmes de fichiers ZFS" à la page 213. Vous pouvez allouer une certaine quantité d'espace disque à un système de fichiers à l'aide des réservations. Pour obtenir des informations

sur les réservations, reportez-vous à la rubrique "Définition de réservations sur les systèmes de fichiers ZFS" à la page 216. Ce modèle est très similaire au modèle NFS dans lequel plusieurs répertoires sont montés à partir du même système de fichiers (par exemple : `/home`).

Toutes les métadonnées dans ZFS sont allouées dynamiquement. La plupart des autres systèmes de fichiers pré-allouent une grande partie de leurs métadonnées. Par conséquent, lors de la création du système de fichiers, ces métadonnées ont besoin d'une partie de l'espace disque. En outre, en raison de ce comportement, le nombre total de fichiers pris en charge par le système de fichiers est prédéterminé. Dans la mesure où ZFS alloue les métadonnées lorsqu'il en a besoin, aucun coût d'espace initial n'est requis et le nombre de fichiers n'est limité que par l'espace disponible. Dans le cas de ZFS, la sortie de la commande `df -g` ne s'interprète pas de la même manière que pour les autres systèmes de fichiers. Le nombre de fichiers (`total files`) indiqué n'est qu'une estimation basée sur la quantité de stockage disponible dans le pool.

ZFS est un système de fichiers transactionnel. La plupart des modifications apportées au système de fichiers sont rassemblées en groupes de transaction et validées sur le disque de façon asynchrone. Tant que ces modifications ne sont pas validées sur le disque, elles sont considérées comme des *modifications en attente*. La quantité d'espace disque utilisé disponible et référencé par un fichier ou un système de fichiers ne tient pas compte des modifications en attente. Ces modifications sont généralement prises en compte au bout de quelques secondes. Même si vous validez une modification apportée au disque avec la commande `fsync(3c)` ou `O_SYNC`, les informations relatives à l'utilisation d'espace disque ne sont pas automatiquement mises à jour.

Sur un système de fichiers UFS, la commande `du` indique la taille des blocs de données au sein du fichier. Sur un système de fichiers ZFS, la commande `du` indique la taille réelle du fichier, telle qu'elle est stockée sur le disque. La taille prend en compte les métadonnées et la compression. Ces informations vous aident à déterminer l'espace supplémentaire dont vous disposerez si vous supprimez un fichier donné. Par conséquent, même lorsque la compression est désactivée, vous obtenez des résultats différents entre ZFS et UFS.

Lorsque vous comparez la consommation d'espace renvoyée par la commande `df` avec celle renvoyée par la commande `zfs list`, n'oubliez pas que `df` indique la taille du pool et pas seulement la taille des systèmes de fichiers. En outre, `df` ne reconnaît pas les systèmes de fichiers descendants ni ne détecte la présence d'instantanés. Si des propriétés ZFS telles que la compression et les quotas sont définies sur les systèmes de fichiers, le rapprochement de la consommation d'espace renvoyée par `df` peut s'avérer difficile.

Considérez les scénarios suivants qui peuvent également avoir un impact sur la consommation d'espace signalée :

- Pour les fichiers de volume supérieur à `recordsize`, le dernier bloc du fichier est généralement à moitié plein. Lorsque `recordsize` est défini par défaut sur 128 Ko, environ 64 Ko sont perdus par fichier, ce qui peut avoir un impact considérable. L'intégration de RFE 6812608 permet de remédier à ce problème. Une solution de contournement consiste à

activer la compression. Même si vos données sont déjà compressées, la partie non utilisée du dernier bloc sera remplie de zéros et sera compressée sans difficulté.

- Sur un pool RAIDZ-2, chaque bloc consomme au moins 2 secteurs (par blocs de 512 octets) d'informations de parité. L'espace utilisé par les informations de parité n'est pas signalé ; toutefois, il peut varier et représenter un pourcentage beaucoup plus élevé pour les blocs de petite taille, si bien qu'il peut avoir une incidence sur les valeurs d'espace renvoyées. L'impact est plus important lorsque `recordsize` est défini sur 512 octets, où chaque bloc logique de 512 octets consomme 1,5 Ko (3 fois l'espace). Quelles que soient les données stockées, si une utilisation efficace de l'espace est primordiale, il est recommandé de conserver la valeur par défaut de `recordsize` (128 KB) et d'activer la compression (sur la valeur par défaut `lzjb`).
- La commande `df` n'a pas connaissance des données de fichiers dédupliquées.

Comportement d'espace saturé

La création d'instantanés de systèmes de fichiers est peu coûteuse et facile dans ZFS. Les instantanés sont communs à la plupart des environnements ZFS. Pour plus d'informations sur les instantanés ZFS, reportez-vous au Chapitre 6, “Utilisation des instantanés et des clones ZFS Oracle Solaris”.

La présence d'instantanés peut entraîner des comportements inattendus lors des tentatives de libération d'espace disque. En règle générale, si vous disposez des autorisations adéquates, vous pouvez supprimer un fichier d'un système de fichiers plein, ce qui entraîne une augmentation de la quantité d'espace disque disponible dans le système de fichiers. Cependant, si le fichier à supprimer existe dans un instantané du système de fichiers, sa suppression ne libère pas d'espace disque. Les blocs utilisés par le fichier continuent à être référencés à partir de l'instantané.

Par conséquent, la suppression du fichier peut occuper davantage d'espace disque car une nouvelle version du répertoire doit être créée afin de refléter le nouvel état de l'espace de noms. En raison de ce comportement, une erreur `ENOSPC` ou `EDQUOT` inattendue peut se produire lorsque vous tentez de supprimer un fichier.

Montage de système de fichiers ZFS

Le système de fichiers ZFS réduit la complexité et facilite l'administration. Par exemple, avec des systèmes de fichiers standard, vous devez modifier le fichier `/etc/vfstab` à chaque fois que vous ajoutez un système de fichiers. Avec ZFS, cela n'est plus nécessaire, grâce au montage et démontage automatique en fonction des propriétés du système de fichiers. Vous n'avez pas besoin de gérer les entrées ZFS dans le fichier `/etc/vfstab`.

Pour plus d'informations sur le montage et le partage des systèmes de fichiers ZFS, reportez-vous à la section “Montage de système de fichiers ZFS” à la page 205.

Gestion de volumes classique

Comme décrit à la section "Stockage ZFS mis en pool" à la page 26, ZFS élimine la nécessité d'un gestionnaire de volume séparé. ZFS opérant sur des périphériques bruts, il est possible de créer un pool de stockage composé de volumes logiques logiciels ou matériels. Cette configuration est déconseillée, car ZFS fonctionne mieux avec des périphériques bruts physiques. L'utilisation de volumes logiques peut avoir un impact négatif sur les performances, la fiabilité, voire les deux, et doit de ce fait être évitée.

Modèle ACL Solaris basé sur NFSv4

Les versions précédentes du système d'exploitation Solaris assuraient la prise en charge d'une implémentation ACL reposant principalement sur la spécification d'ACL POSIX-draft. Les ACL POSIX-draft sont utilisées pour protéger des fichiers UFS. Un nouveau modèle ACL basé sur la spécification NFSv4 est utilisé pour protéger les fichiers ZFS.

Les principales différences présentées par le nouveau modèle ACL Solaris sont les suivantes :

- Le modèle est basé sur la spécification NFSv4 et similaire aux ACL de type Windows NT.
- Ce modèle fournit un jeu d'autorisations d'accès plus détaillé.
- Les ACL sont définies et affichées avec les commandes `chmod` et `ls` plutôt qu'avec les commandes `setfacl` et `getfacl` .
- Une sémantique d'héritage plus riche désigne la manière dont les privilèges d'accès sont appliqués d'un répertoire à un sous-répertoire, et ainsi de suite.

Pour plus d'informations sur l'utilisation des ACL avec des fichiers ZFS, reportez-vous au Chapitre 7, "Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS".

◆ ◆ ◆ CHAPITRE 2

Mise en route d'Oracle Solaris ZFS

Ce chapitre fournit des instructions détaillées sur la configuration de base d'Oracle Solaris ZFS. Il offre une vision globale du fonctionnement des commandes ZFS et explique les méthodes de création de pools et de systèmes de fichiers de base. Ce chapitre ne constitue pas une présentation exhaustive. Pour des informations plus détaillées, reportez-vous aux autres chapitres, comme indiqué.

Ce chapitre contient les sections suivantes :

- "Profils de droits ZFS" à la page 37
- "Exigences et recommandations en matière de matériel et de logiciel ZFS " à la page 38
- "Création d'un système de fichiers ZFS de base" à la page 38
- "Création d'un pool de stockage ZFS de base" à la page 39
- "Création d'une hiérarchie de systèmes de fichiers ZFS" à la page 41

Profils de droits ZFS

Si vous souhaitez effectuer des tâches de gestion ZFS sans utiliser le compte superutilisateur (root), vous pouvez prendre un rôle disposant de l'une des propriétés suivantes afin d'effectuer des tâches d'administration ZFS :

- Gestion de stockage ZFS : permet de créer, détruire, manipuler des périphériques au sein d'un pool de stockage ZFS.
- Gestion de système de fichiers ZFS : spécifie les autorisations de création, de destruction et de modification des systèmes de fichiers ZFS.

Pour de plus amples informations sur la création ou l'assignation de rôles, reportez-vous au *System Administration Guide: Security Services*.

Outre les rôles RBAC permettant de gérer les systèmes de fichiers ZFS, vous pouvez également vous servir de l'administration déléguée de ZFS pour effectuer des tâches d'administration ZFS distribuée. Pour plus d'informations, reportez-vous au Chapitre 8, "Administration déléguée de ZFS dans Oracle Solaris".

Exigences et recommandations en matière de matériel et de logiciel ZFS

Avant d'utiliser le logiciel ZFS, passez en revue les exigences et recommandations matérielles et logicielles suivantes :

- Utilisez un système SPARC ou un système x86 exécutant une version d'Oracle Solaris prise en charge.
- L'espace disque minimum requis pour un pool de stockage est de 64 Mo. La taille minimale du disque est de 128 Mo.
- Pour des performances ZFS optimales, évaluez la taille de mémoire requise en fonction de votre charge de travail.
- Si vous créez une configuration de pool mis en miroir, utilisez plusieurs contrôleurs.

Création d'un système de fichiers ZFS de base

L'administration de ZFS a été conçue dans un but de simplicité. L'un des objectifs principaux est de réduire le nombre de commandes nécessaires à la création d'un système de fichiers utilisable. Par exemple, lors de la création d'un pool, un système de fichiers ZFS est automatiquement créé et monté.

L'exemple suivant illustre la création d'un pool de stockage à miroir simple appelé `tank` et d'un système de fichiers ZFS appelé `tank`, en une seule commande. Supposons que l'intégralité des disques `/dev/dsk/c1t0d0` et `/dev/dsk/c2t0d0` puissent être utilisés.

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

Pour plus d'informations sur les configurations redondantes de pools ZFS, reportez-vous à la section “Fonctions de réplication d'un pool de stockage ZFS” à la page 49.

Le nouveau système de fichiers ZFS, `tank`, peut utiliser autant d'espace disque disponible que nécessaire et est monté automatiquement sur `/tank`.

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

Au sein d'un pool, vous souhaiterez probablement créer des systèmes de fichiers supplémentaires. Les systèmes de fichiers fournissent des points d'administration qui permettent de gérer différents jeux de données au sein du même pool.

L'exemple illustre la création d'un système de fichiers nommé `fs` dans le pool de stockage `tank`.

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

Le nouveau système de fichiers ZFS, `tank/fs`, peut utiliser autant d'espace disque disponible que nécessaire et est monté automatiquement sur `/tank/fs`.

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

Généralement, vous souhaitez créer et organiser une hiérarchie de systèmes de fichiers correspondant à des besoins spécifiques en matière d'organisation. Pour de plus amples informations sur la création d'une hiérarchie de systèmes de fichiers ZFS, reportez-vous à la section "Création d'une hiérarchie de systèmes de fichiers ZFS" à la page 41.

Création d'un pool de stockage ZFS de base

L'exemple suivant illustre la simplicité de ZFS. Vous trouverez dans la suite de cette section un exemple plus complet, similaire à ce qui pourrait exister dans votre environnement. Les premières tâches consistent à identifier les besoins en matière de stockage et à créer un pool de stockage. Le pool décrit les caractéristiques physiques du stockage et doit être créé préalablement à tout système de fichiers.

▼ Identification des exigences de stockage du pool de stockage ZFS

1 Déterminez les périphériques disponibles pour le pool de stockage.

Avant de créer un pool de stockage, vous devez définir les périphériques à utiliser pour stocker les données. Ces périphériques doivent être des disques de 128 Mo minimum et ne doivent pas être en cours d'utilisation par d'autres parties du système d'exploitation. Il peut s'agir de tranches individuelles d'un disque préformaté ou de disques entiers formatés par ZFS sous forme d'une seule grande tranche.

Pour l'exemple de stockage utilisé dans la section "Création d'un pool de stockage ZFS" à la page 40, partez du principe que les disques entiers `/dev/dsk/c1t0d0` et `/dev/dsk/c2t0d0` sont disponibles.

Pour de plus amples informations sur les disques, leur utilisation et leur étiquetage, reportez-vous à la section "Utilisation de disques dans un pool de stockage ZFS" à la page 45.

2 Sélectionnez la réplication de données.

Le système de fichiers ZFS assure la prise en charge de plusieurs types de réplication de données. Cela permet de déterminer les types de panne matérielle supportés par le pool. ZFS assure la prise en charge des configurations non redondantes (entrelacées), ainsi que la mise en miroir et RAID-Z (une variante de RAID-5).

Pour l'exemple de stockage utilisé dans la section "Création d'un pool de stockage ZFS" à la page 40 utilise la mise en miroir de base de deux disques disponibles.

Pour de plus amples informations sur les fonctions de réplication ZFS, reportez-vous à la section "Fonctions de réplication d'un pool de stockage ZFS" à la page 49.

▼ Création d'un pool de stockage ZFS

1 **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent avec un profil de droits ZFS adéquat.**

Pour de plus amples informations sur les profils de droits ZFS, reportez-vous à la section "Profils de droits ZFS" à la page 37.

2 **Assignez un nom au pool de stockage.**

Le nom sert à identifier le pool de stockage lorsque vous exécutez les commandes `zpool` et `zfs`. Entrez le nom de votre choix, mais celui-ci doit respecter les conventions d'attribution de nom définies dans la section "Exigences d'attribution de noms de composants ZFS" à la page 31.

3 **Créez le pool.**

Par exemple, la commande suivante crée un pool mis en miroir nommé `tank` :

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

Si des périphériques contiennent un autre système de fichiers ou sont en cours d'utilisation, la commande ne peut pas créer le pool.

Pour plus d'informations sur la création de pools de stockage, reportez-vous à la section "Création de pools de stockage ZFS" à la page 52. Pour plus d'informations sur la détection de l'utilisation de périphériques, reportez-vous à la section "Détection des périphériques utilisés" à la page 60.

4 **Affichez les résultats.**

Vous pouvez déterminer si votre pool a été correctement créé à l'aide de la commande `zpool list`.

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

Pour de plus amples informations sur la vérification de l'état de pool, reportez-vous à la section "Requête d'état de pool de stockage ZFS" à la page 88.

Création d'une hiérarchie de systèmes de fichiers ZFS

Une fois le pool de stockage, vous pouvez créer la hiérarchie du système de fichiers. Les hiérarchies sont des mécanismes d'organisation des informations à la fois simples et puissants. Elles sont connues de toute personne ayant utilisé un système de fichiers.

ZFS permet d'organiser en hiérarchies les systèmes de fichiers. Chaque système de cette hiérarchie ne compte qu'un seul parent. Le root de la hiérarchie correspond toujours au nom du pool. ZFS exploite cette hiérarchie en assurant la prise en charge de l'héritage de propriétés. Ainsi, vous pouvez définir les propriétés communes rapidement et facilement dans des arborescences représentant l'intégralité des systèmes de fichiers.

▼ Détermination de la hiérarchie du système de fichiers ZFS

1 Choisissez la granularité du système de fichiers.

Les systèmes de fichiers ZFS sont le point central d'administration. Ils sont légers et se créent facilement. Pour ce faire, nous vous recommandons d'établir un système de fichiers par utilisateur ou par projet car cela permet de contrôler les propriétés, les instantanés et les sauvegardes par utilisateur ou par projet.

Deux systèmes de fichiers ZFS, `jeff` et `bill`, sont créés à la section "Création de systèmes de fichiers ZFS" à la page 42.

Pour plus d'informations sur la gestion des systèmes de fichiers, reportez-vous au Chapitre 5, "Gestion des systèmes de fichiers Oracle Solaris ZFS".

2 Regroupez les systèmes de fichiers similaires.

ZFS permet d'organiser les systèmes de fichiers en hiérarchie, pour regrouper les systèmes de fichiers similaires. Ce modèle fournit un point d'administration central pour le contrôle des propriétés et l'administration de systèmes de fichiers. Il est recommandé de créer les systèmes de fichiers similaires sous un nom commun.

Dans l'exemple de la section "Création de systèmes de fichiers ZFS" à la page 42, les deux systèmes de fichiers sont placés sous un système de fichiers appelé `home`.

3 Choisissez les propriétés du système de fichiers.

La plupart des caractéristiques de systèmes de fichiers se contrôlent à l'aide de propriétés. Ces propriétés assurent le contrôle de divers comportements, y compris l'emplacement de montage des systèmes de fichiers, leur méthode de partage, l'utilisation de la compression et l'activation des quotas.

Dans l'exemple de la section "Création de systèmes de fichiers ZFS" à la page 42, tous les répertoires personnels sont montés dans `/export/zfs/` *user*. Ils sont partagés à l'aide de NFS et la compression est activée. De plus, un quota de 10 Go est appliqué pour l'utilisateur `jeff`.

Pour de plus amples informations sur les propriétés, reportez-vous à la section "Présentation des propriétés ZFS" à la page 183.

▼ Création de systèmes de fichiers ZFS

1 **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent avec un profil de droits ZFS adéquat.**

Pour de plus amples informations sur les profils de droits ZFS, reportez-vous à la section "Profils de droits ZFS" à la page 37.

2 **Créez la hiérarchie souhaitée.**

Dans cet exemple, un système de fichiers agissant en tant que conteneur de systèmes de fichiers individuels est créé.

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

3 **Définissez les propriétés héritées.**

Une fois la hiérarchie du système de fichiers établie, définissez toute propriété destinée à être partagée par l'ensemble des utilisateurs :

```
# 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
```

Il est possible de définir les propriétés du système de fichiers lors de la création de ce dernier. Par exemple :

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

Pour plus d'informations sur les propriétés et l'héritage des propriétés, reportez-vous à la section "Présentation des propriétés ZFS" à la page 183.

Ensuite, les systèmes de fichiers sont regroupés sous le système de fichiers `home` dans le pool `tank`.

4 **Créez les systèmes de fichiers individuels.**

Il est possible que les systèmes de fichiers aient été créés et que leurs propriétés aient ensuite été modifiées au niveau `home`. Vous pouvez modifier les propriétés de manière dynamique lorsque les systèmes de fichiers sont en cours d'utilisation.

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

Les valeurs de propriétés de ces systèmes de fichiers sont héritées de leur parent. Elles sont donc montées sur `/export/zfs/` *user* et partagées via NFS. Il est inutile de modifier le fichier `/etc/vfstab` ou `/etc/dfs/dfstab`.

Pour de plus amples informations sur les systèmes de fichiers, reportez-vous à la section "Création d'un système de fichiers ZFS" à la page 180.

Pour plus d'informations sur le montage et le partage de systèmes de fichiers, reportez-vous à la section "Montage de système de fichiers ZFS" à la page 205.

5 Définissez les propriétés spécifiques au système.

Dans cet exemple, l'utilisateur `jeff` se voit assigner un quota de 10 Go. Cette propriété place une limite sur la quantité d'espace qu'il peut utiliser, indépendamment de l'espace disponible dans le pool.

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

6 Affichez les résultats.

La commande `zfs list` permet de visualiser les informations disponibles sur le système de fichiers :

```
# 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
```

Notez que l'utilisateur `jeff` dispose d'uniquement 10 Go d'espace disponible, tandis que l'utilisateur `bill` peut utiliser la totalité du pool (67 Go).

Pour de plus amples informations sur la visualisation de l'état du système de fichiers, reportez-vous à la section "Envoi de requêtes sur les informations des systèmes de fichiers ZFS" à la page 198.

Pour de plus amples informations sur l'utilisation et le calcul de l'espace disque, reportez-vous à la section "Comptabilisation de l'espace disque ZFS" à la page 32.

◆ ◆ ◆ CHAPITRE 3

Gestion des pools de stockage Oracle Solaris ZFS

Ce chapitre explique comment créer et administrer des pools de stockage dans Oracle Solaris ZFS.

Ce chapitre contient les sections suivantes :

- "Composants d'un pool de stockage ZFS" à la page 45
- "Fonctions de réplication d'un pool de stockage ZFS" à la page 49
- "Création et destruction de pools de stockage ZFS" à la page 52
- "Gestion de périphériques dans un pool de stockage ZFS" à la page 64
- "Gestion des propriétés de pool de stockage ZFS" à la page 85
- "Requête d'état de pool de stockage ZFS" à la page 88
- "Migration de pools de stockage ZFS" à la page 100
- "Mise à niveau de pools de stockage ZFS" à la page 109

Composants d'un pool de stockage ZFS

Les sections ci-dessous contiennent des informations détaillées sur les composants de pools de stockage suivants :

- "Utilisation de disques dans un pool de stockage ZFS" à la page 45
- "Utilisation de tranches dans un pool de stockage ZFS" à la page 46
- "Utilisation de fichiers dans un pool de stockage ZFS" à la page 48

Utilisation de disques dans un pool de stockage ZFS

Le composant le plus élémentaire d'un pool de stockage est le stockage physique. Le stockage physique peut être constitué de tout périphérique en mode bloc d'une taille supérieure à 128 Mo. En règle générale, ce périphérique est un disque dur visible pour le système dans le répertoire `/dev/dsk`.

Un disque entier (`c1t0d0`) ou une tranche individuelle (`c0t0d0s7`) peuvent constituer un périphérique de stockage. La manière d'opérer recommandée consiste à utiliser un disque entier. Dans ce cas, il est inutile de formater spécifiquement le disque. ZFS formate le disque à l'aide d'une étiquette EFI de façon à ce qu'il contienne une grande tranche unique. Utilisé de cette façon, le tableau de partition affiché par la commande `format` s'affiche comme suit :

```
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
```

Tenez compte des points suivants lorsque vous utilisez des disques entiers dans vos pools de stockage ZFS :

- Lorsqu'un disque entier est utilisé, le disque est généralement nommé à l'aide de la convention de nommage `/dev/dsk/cNtNdN`. Certains pilotes tiers suivent une convention de nom différente ou placent les disques à un endroit autre que le répertoire `/dev/dsk`. Pour utiliser ces disques, vous devez les étiqueter manuellement et fournir une tranche à ZFS.
- Sur les systèmes x86, le disque doit avoir une partition `fdisk` Solaris valide. Pour plus d'informations sur la création et la modification d'une partition `fdisk` Solaris, reportez-vous à la section "Setting Up Disks for ZFS File Systems (Task Map)" du manuel *System Administration Guide: Devices and File Systems*.
- ZFS applique une étiquette EFI lorsque vous créez un pool de stockage avec des disques entiers. Pour plus d'informations sur les étiquettes EFI, reportez-vous à la section "EFI (GPT) Disk Label" du manuel *System Administration Guide: Devices and File Systems*.

Vous pouvez spécifier les disques soit en utilisant le chemin complet (`/dev/dsk/c1t0d0`, par exemple) ou un nom abrégé composé du nom du périphérique dans le répertoire `/dev/dsk` (`c1t0d0`, par exemple). Les exemples suivants constituent des noms de disques valides :

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

Utilisation de tranches dans un pool de stockage ZFS

Les disques peuvent être étiquetés avec une étiquette Solaris VTOC (SMI) quand vous créez un pool de stockage avec une tranche de disque, mais l'utilisation de tranches de disque pour un pool n'est pas recommandée car leur gestion est plus difficile.

Sur un système SPARC, un disque de 72 Go dispose de 68 Go d'espace utilisable situé dans la tranche 0, comme illustré dans la sortie `format` suivante :

```
# 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
```

Sur un système x86, un disque de 72 Go dispose de 68 Go d'espace disque utilisable situé dans la tranche 0, comme illustré dans la sortie `format` suivante : Une petite quantité d'informations d'initialisation est contenue dans la tranche 8. La tranche 8 ne nécessite aucune administration et ne peut pas être modifiée.

```
# 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
```

Une partition `fdisk` existe également sur les systèmes x86. Une partition `fdisk` est représentée par un nom de périphérique `/dev/dsk/cN[tN]dNpN` et fait office de conteneur pour les tranches disponibles du disque. N'utilisez pas de périphérique `cN[tN]dNpN` pour un composant de pool de stockage ZFS car cette configuration n'est ni testée ni prise en charge.

Utilisation de fichiers dans un pool de stockage ZFS

ZFS permet également d'utiliser des fichiers en tant que périphériques virtuels dans le pool de stockage. Cette fonction est destinée principalement aux tests et à des essais simples, et non pas à être utilisée dans un contexte de production.

- Si vous créez un pool ZFS sauvegardé par des fichiers dans un système de fichiers UFS, vous vous basez implicitement sur UFS pour la garantie de l'exactitude et de la synchronisation de la sémantique.
- Si vous créez un pool ZFS à partir de fichiers ou de volumes créés sur un autre pool ZFS, le système peut générer un interblocage ou paniquer.

Cependant, les fichiers peuvent s'avérer utiles lorsque vous employez ZFS pour la première fois ou en cas de configuration complexe, lorsque les périphériques physiques présents ne sont pas suffisants. Tous les fichiers doivent être spécifiés avec leur chemin complet et leur taille doit être de 64 Mo minimum.

Remarques relatives aux pools de stockage ZFS

Tenez compte des points suivants lors de la création et de la gestion de pools de stockage ZFS.

- L'utilisation de disques physiques constitue la méthode de création de pools de stockage ZFS la plus simple. Les configurations ZFS deviennent de plus en plus complexes, en termes de gestion, de fiabilité et de performance. Lorsque vous construisez des pools à partir de tranches de disques, de LUN dans des baies RAID matérielles ou de volumes présentés par des gestionnaires de volume basés sur des logiciels. Les considérations suivantes peuvent vous aider à configurer ZFS avec d'autres solutions de stockage matérielles ou logicielles :
  - Si vous élaborez une configuration ZFS sur des LUN à partir de baies RAID matérielles, vous devez comprendre la relation entre les fonctionnalités de redondance ZFS et les fonctionnalités de redondance proposées par la baie. Certaines configurations peuvent fournir une redondance et des performances adéquates, mais d'autres non.
  - Vous pouvez construire des périphériques logiques pour ZFS à l'aide de volumes présentés par des gestionnaires de volumes logiciels tels que Solaris Volume Manager (SVM) ou Veritas Volume Manager (VxVM). Ces configurations sont cependant déconseillées. Même si le système de fichiers ZFS fonctionne correctement sur ces périphériques, il se peut que les performances ne soient pas optimales.

    Pour plus d'informations sur les recommandations relatives aux pools de stockage, reportez-vous au Chapitre 11, “Pratiques recommandées pour Oracle Solaris ZFS”.
- Les disques sont identifiés par leur chemin et par l'ID de leur périphérique, s'il est disponible. Pour les systèmes sur lesquels les informations de l'ID du périphérique sont disponibles, cette méthode d'identification permet de reconfigurer les périphériques sans mettre à jour ZFS. Etant donné que la génération et la gestion d'ID de périphérique peuvent varier d'un système à l'autre, vous devez commencer par exporter le pool avant tout

déplacement de périphériques, par exemple, le déplacement d'un disque d'un contrôleur à un autre. Un événement système, tel que la mise à jour du microprogramme ou toute autre modification apportée au matériel, peut modifier les ID de périphérique du pool de stockage ZFS, ce qui peut entraîner l'indisponibilité des périphériques.

Fonctions de réplication d'un pool de stockage ZFS

Le système de fichiers ZFS offre une redondance des données, ainsi que des propriétés d'auto-rétablissement dans des configurations RAID-Z ou mises en miroir.

- "Configuration de pool de stockage mis en miroir" à la page 49
- "Configuration de pool de stockage RAID-Z" à la page 49
- "Données d'autorétablissement dans une configuration redondante" à la page 51
- "Entrelacement dynamique dans un pool de stockage" à la page 51
- "Pool de stockage ZFS hybride" à la page 51

Configuration de pool de stockage mis en miroir

Une configuration de pool de stockage en miroir requiert deux disques minimum, situés de préférence dans des contrôleurs séparés. Vous pouvez utiliser un grand nombre de disques dans une configuration en miroir. En outre, vous pouvez créer plusieurs miroirs dans chaque pool. Conceptuellement, une configuration en miroir de base devrait ressembler à ce qui suit :

```
mirror c1t0d0 c2t0d0
```

Conceptuellement, une configuration en miroir plus complexe devrait ressembler à ce qui suit :

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

Pour obtenir des informations sur les pools de stockage mis en miroir, reportez-vous à la section "Création d'un pool de stockage mis en miroir" à la page 52.

Configuration de pool de stockage RAID-Z

En plus d'une configuration en miroir de pool de stockage, ZFS fournit une configuration RAID-Z disposant d'une tolérance de pannes à parité simple, double ou triple. Une configuration RAID-Z à parité simple (`raidz` ou `raidz1`) équivaut à une configuration RAID-5. Une configuration RAID-Z à double parité (`raidz2`) est similaire à une configuration RAID-6.

Pour plus d'informations sur la fonction RAIDZ-3 (`raidz3`), consultez le blog suivant :

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

Tous les algorithmes similaires à RAID-5 (RAID-4, RAID-6, RDP et EVEN-ODD, par exemple) peuvent souffrir d'un problème connu sous le nom de *RAID-5 write hole*, ou trou d'écriture de RAID-5. Si seule une partie d'un entrelacement RAID-5 est écrite, et qu'une perte d'alimentation se produit avant que tous les blocs aient été écrits sur le disque, la parité n'est pas synchronisée avec les données, et est par conséquent inutile à tout jamais (à moins qu'elle ne soit écrasée ultérieurement par une écriture d'entrelacement total). Dans RAID-Z, ZFS utilise des entrelacements RAID de largeur variable pour que toutes les écritures correspondent à des entrelacements entiers. Cette conception n'est possible que parce que ZFS intègre le système de fichiers et la gestion de périphérique de telle façon que les métadonnées du système de fichiers disposent de suffisamment d'informations sur le modèle de redondance de données pour gérer les entrelacements RAID de largeur variable. RAID-Z est la première solution au monde pour le trou d'écriture de RAID-5.

Une configuration RAID-Z avec N disques de taille X et des disques de parité P présente une contenance d'environ (N-P)*X octets et peut supporter la panne d'un ou de plusieurs périphériques P avant que l'intégrité des données ne soit compromise. Vous devez disposer d'au moins deux disques pour une configuration RAID-Z à parité simple et d'au moins trois disques pour une configuration RAID-Z à double parité, et ainsi de suite. Par exemple, si vous disposez de trois disques pour une configuration RAID-Z à parité simple, les données de parité occupent un espace disque égal à l'un des trois disques. Dans le cas contraire, aucun matériel spécifique n'est requis pour la création d'une configuration RAID-Z.

Conceptuellement, une configuration RAID-Z à trois disques serait similaire à ce qui suit :

```
raidz c1t0d0 c2t0d0 c3t0d0
```

Conceptuellement, une configuration RAID-Z plus complexe devrait ressembler à ce qui suit :

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

Si vous créez une configuration RAID-Z avec un nombre important de disques, vous pouvez scinder les disques en plusieurs groupes. Par exemple, il est recommandé d'utiliser une configuration RAID-Z composée de 14 disques au lieu de la scinder en 2 groupes de 7 disques. Les configurations RAID-Z disposant de groupements de moins de 10 disques devraient présenter de meilleures performances.

Pour obtenir des informations sur les pools de stockage RAID-Z, reportez-vous à la section “Création d'un pool de stockage RAID-Z” à la page 54.

Pour obtenir des informations supplémentaires afin de choisir une configuration en miroir ou une configuration RAID-Z en fonction de considérations de performances et d'espace disque, consultez le blog suivant :

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

Pour plus d'informations sur les recommandations relatives aux pools de stockage RAID-Z, reportez-vous au Chapitre 11, “Pratiques recommandées pour Oracle Solaris ZFS”.

Pool de stockage ZFS hybride

Le pool de stockage ZFS hybride est disponible dans la gamme de produits Oracle Sun Storage 7000. Il s'agit d'un pool de stockage spécial combinant de la RAM dynamique, des disques électroniques et des disques durs, qui permet d'améliorer les performances et d'augmenter la capacité, tout en réduisant la consommation électrique. Grâce à l'interface de gestion de ce produit, vous pouvez sélectionner la configuration de redondance ZFS du pool de stockage et gérer facilement d'autres options de configuration.

Pour plus d'informations sur ce produit, reportez-vous au *Sun Storage Unified Storage System Administration Guide*.

Données d'autorétablissement dans une configuration redondante

Le système de fichiers ZFS fournit des données d'auto-rétablissement dans une configuration RAID-Z ou mise en miroir.

Lorsqu'un bloc de données endommagé est détecté, ZFS récupère les données correctes à partir d'une copie redondante et de plus, répare les données incorrectes en les remplaçant par celles de la copie.

Entrelacement dynamique dans un pool de stockage

Le système de fichiers ZFS entrelace de façon dynamique les données de tous les périphériques virtuels de niveau supérieur. Le choix de l'emplacement des données est effectué lors de l'écriture ; ainsi, aucun entrelacement de largeur fixe n'est créé lors de l'allocation.

Lorsque de nouveaux périphériques virtuels sont ajoutés à un pool, ZFS attribue graduellement les données au nouveau périphérique afin de maintenir les performances et les stratégies d'allocation d'espace disque. Chaque périphérique virtuel peut également être constitué d'un miroir ou d'un périphérique RAID-Z contenant d'autres périphériques de disques ou d'autres fichiers. Cette configuration vous offre un contrôle flexible des caractéristiques par défaut du pool. Par exemple, vous pouvez créer les configurations suivantes à partir de quatre disques :

- Quatre disques utilisant l'entrelacement dynamique
- Une configuration RAID-Z à quatre directions
- Deux miroirs bidirectionnels utilisant l'entrelacement dynamique

Même si le système de fichiers ZFS prend en charge différents types de périphériques virtuels au sein du même pool, cette pratique n'est pas recommandée. Vous pouvez par exemple créer un pool avec un miroir bidirectionnel et une configuration RAID-Z à trois directions. Cependant, le niveau de tolérance de pannes est aussi bon que le pire périphérique virtuel (RAID-Z dans ce cas). Nous vous recommandons d'utiliser des périphériques virtuels de niveau supérieur du même type avec le même niveau de redondance pour chaque périphérique.

Création et destruction de pools de stockage ZFS

Les sections suivantes illustrent différents scénarios de création et de destruction de pools de stockage ZFS :

- "Création de pools de stockage ZFS" à la page 52
- "Affichage des informations d'un périphérique virtuel de pool de stockage" à la page 59
- "Gestion d'erreurs de création de pools de stockage ZFS" à la page 60
- "Destruction de pools de stockage ZFS" à la page 63

La création et la destruction de pools est rapide et facile. Cependant, ces opérations doivent être réalisées avec prudence. Des vérifications sont effectuées pour éviter une utilisation de périphériques déjà utilisés dans un nouveau pool, mais ZFS n'est pas systématiquement en mesure de savoir si un périphérique est déjà en cours d'utilisation. Il est plus facile de détruire un pool que d'en créer un. Utilisez la commande `zpool destroy` avec précaution. L'exécution de cette commande simple a des conséquences considérables.

Création de pools de stockage ZFS

Pour créer un pool de stockage, exécutez la commande `zpool create`. Cette commande prend un nom de pool et un nombre illimité de périphériques virtuels en tant qu'arguments. Le nom de pool doit se conformer aux conventions d'attribution de noms décrites à la section "Exigences d'attribution de noms de composants ZFS" à la page 31.

Création d'un pool de stockage de base

La commande suivante crée un pool appelé `tank` et composé des disques `c1t0d0` et `c1t1d0`:

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

Ces noms de périphériques représentant les disques entiers se trouvent dans le répertoire `/dev/dsk` et ont été étiquetés de façon adéquate par ZFS afin de contenir une tranche unique de grande taille. Les données sont entrelacées de façon dynamique sur les deux disques.

Création d'un pool de stockage mis en miroir

Pour créer un pool mis en miroir, utilisez le mot-clé `mirror` suivi du nombre de périphériques de stockage que doit contenir le miroir. Pour spécifier plusieurs miroirs, répétez le mot-clé `mirror` dans la ligne de commande. La commande suivante crée un pool avec deux miroirs bidirectionnels :

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

Le second mot-clé `mirror` indique qu'un nouveau périphérique virtuel de niveau supérieur est spécifié. Les données sont dynamiquement entrelacées sur les deux miroirs, ce qui les rend redondantes sur chaque disque.

Pour plus d'informations sur les configurations en miroir recommandées, reportez-vous au Chapitre 11, “Pratiques recommandées pour Oracle Solaris ZFS”.

Actuellement, les opérations suivantes sont prises en charge dans une configuration ZFS en miroir :

- Ajout d'un autre jeu de disques comme périphérique virtuel (vdev) supplémentaire de niveau supérieur à une configuration en miroir existante. Pour plus d'informations, reportez-vous à la rubrique “Ajout de périphériques à un pool de stockage” à la page 64.
- Connexion de disques supplémentaires à une configuration en miroir existante ou connexion de disques supplémentaires à une configuration non répliquée pour créer une configuration en miroir. Pour plus d'informations, reportez-vous à la section “Connexion et séparation de périphériques dans un pool de stockage ” à la page 69.
- Remplacement d'un ou de plusieurs disques dans une configuration en miroir existante, à condition que les disques de remplacement soient d'une taille supérieure ou égale à celle du périphérique remplacé. Pour plus d'informations, reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 77.
- Retrait d'un ou de plusieurs disques dans une configuration en miroir, à condition que les périphériques restants procurent la redondance qui convient à la configuration. Pour plus d'informations, reportez-vous à la section “Connexion et séparation de périphériques dans un pool de stockage ” à la page 69.
- Scission d'une configuration mise en miroir en déconnectant l'un des disques en vue de créer un nouveau pool identique. Pour plus d'informations, reportez-vous à la section “Création d'un pool par scission d'un pool de stockage ZFS mis en miroir” à la page 71.

Vous ne pouvez pas forcer la suppression d'un périphérique qui n'est pas un périphérique de rechange, un périphérique de journalisation ou un périphérique de cache d'un pool de stockage mis en miroir.

Création d'un pool root ZFS

Tenez compte des exigences suivantes applicables à la configuration du pool root :

- Le pool root doit être créé sous la forme d'une configuration en miroir ou d'une configuration à disque unique. Vous ne pouvez pas ajouter d'autres disques mis en miroir pour créer plusieurs périphériques virtuels de niveau supérieur à l'aide de la commande `zpool add`. Toutefois, vous pouvez étendre un périphérique virtuel mis en miroir à l'aide de la commande `zpool attach`.
- Les configurations RAID-Z ou entrelacées ne sont pas prises en charge.
- Un pool root ne peut pas avoir de périphérique de journalisation distinct.
- Si vous tentez d'utiliser une configuration non prise en charge pour un pool root, un message tel que le suivant s'affiche :

```
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
```

Pour plus d'informations sur l'installation et l'initialisation d'un système de fichiers root ZFS, reportez-vous Chapitre 4, “Installation et initialisation d'un système de fichiers root ZFS Oracle Solaris”.

Création d'un pool de stockage RAID-Z

La création d'un pool RAID-Z à parité simple est identique à celle d'un pool mis en miroir, à la seule différence que le mot-clé `raidz` ou `raidz1` est utilisé à la place du mot-clé `mirror`. Les exemples suivants illustrent la création d'un pool avec un périphérique RAID-Z unique composé de cinq disques :

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

Cet exemple montre que les disques peuvent être spécifiés à l'aide de leurs noms de périphérique abrégés ou complets. Les deux éléments `/dev/dsk/c5t0d0` et `c5t0d0` font référence au même disque.

Vous pouvez créer une configuration RAID-Z à double ou à triple parité à l'aide du mot-clé `raidz2` ou `raidz3` lors de la création du pool. Par exemple :

```
# 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
```

Actuellement, les opérations suivantes sont prises en charge dans une configuration RAID-Z ZFS :

- Ajout d'un autre jeu de disques comme périphérique virtuel (vdev) supplémentaire de niveau supérieur à une configuration RAID-Z existante. Pour plus d'informations, reportez-vous à la rubrique "Ajout de périphériques à un pool de stockage" à la page 64.
- Remplacement d'un ou de plusieurs disques dans une configuration RAID-Z existante, à condition que les disques de remplacement soient d'une taille supérieure ou égale au celle du périphérique remplacé. Pour plus d'informations, reportez-vous à la section "Remplacement de périphériques dans un pool de stockage" à la page 77.

Actuellement, les opérations suivantes ne sont *pas* prises en charge dans une configuration RAID-Z :

- Connexion d'un disque supplémentaire à une configuration RAID-Z existante.
- Déconnexion d'un disque d'une configuration RAID-Z, sauf lorsque vous déconnectez un disque qui est remplacé par un disque de rechange ou lorsque vous avez besoin de déconnecter un disque de rechange.
- Vous ne pouvez pas forcer la suppression d'un périphérique qui n'est pas un périphérique de journalisation ni de cache à partir d'une configuration RAID-Z. Cette fonction fait l'objet d'une demande d'amélioration.

Pour obtenir des informations supplémentaire, reportez-vous à la section "Configuration de pool de stockage RAID-Z" à la page 49.

Création d'un pool de stockage ZFS avec des périphériques de journalisation

Le journal d'intention ZFS (ZIL) permet de répondre aux exigences de la norme POSIX dans le cadre de transactions synchronisées. Par exemple, les transactions de base de données doivent souvent se trouver sur des périphériques de stockage stables lorsqu'elles sont obtenues à partir d'un appel système. NFS et d'autres applications peuvent également utiliser `fsync()` pour assurer la stabilité des données.

Par défaut, le ZIL est attribué à partir de blocs dans le pool principal. Il est cependant possible d'obtenir de meilleures performances en utilisant des périphériques de journalisation d'intention distincts, notamment une NVRAM ou un disque dédié.

Considérez les points suivants pour déterminer si la configuration d'un périphérique de journalisation ZFS convient à votre environnement :

- Les périphériques de journalisation du ZIL ne sont pas liés aux fichiers journaux de base de données.
- Toute amélioration des performances observée suite à l'implémentation d'un périphérique de journalisation distinct dépend du type de périphérique, de la configuration matérielle du pool et de la charge de travail de l'application. Pour des informations préliminaires sur les performances, consultez le blog suivant :

  http://blogs.oracle.com/perrin/entry/slog_blog_or_blogging_on
- Les périphériques de journalisation peuvent être mis en miroir et leur réplication peut être annulée, mais RAID-Z n'est pas pris en charge pour les périphériques de journalisation.
- Si un périphérique de journalisation distinct n'est pas mis en miroir et que le périphérique contenant le journal échoue, le stockage des blocs de journal retourne sur le pool de stockage.
- Les périphériques de journalisation peuvent être ajoutés, remplacés, connectés, déconnectés, importés et exportés en tant que partie du pool de stockage.
- Vous pouvez connecter un périphérique de journalisation à un périphérique de journalisation existant afin de créer un périphérique mis en miroir. Cette opération est similaire à la connexion d'un périphérique à un pool de stockage qui n'est pas mis en miroir.
- La taille minimale d'un périphérique de journalisation correspond à la taille minimale de chaque périphérique d'un pool, à savoir 64 Mo. La quantité de données en jeu pouvant être stockée sur un périphérique de journalisation est relativement petite. Les blocs de journal sont libérés lorsque la transaction du journal (appel système) est validée.
- La taille maximale d'un périphérique de journalisation doit être approximativement égale à la moitié de la taille de la mémoire physique car il s'agit de la quantité maximale de données en jeu potentielles pouvant être stockée. Si un système dispose par exemple de 16 Go de mémoire physique, considérez une taille maximale de périphérique de journalisation de 8 Go.

Vous pouvez installer un périphérique de journalisation ZFS au moment de la création du pool de stockage ou après sa création.

L'exemple suivant explique comment créer un pool de stockage mis en miroir contenant des périphériques de journalisation mis en miroir :

```
# 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
```

Pour plus d'informations sur la récupération suite à une défaillance de périphérique de journalisation, reportez-vous à l'Exemple 10–2.

Création d'un pool de stockage ZFS avec des périphériques de cache

Les périphériques de cache fournissent une couche de mise en cache supplémentaire entre la mémoire principale et le disque. L'utilisation de périphériques de cache constitue la meilleure amélioration de performances pour les charges de travail de lecture aléatoire constituées principalement de contenu statique.

Vous pouvez créer un pool de stockage avec des périphériques de cache afin de mettre en cache des données de pool de stockage. Par exemple :

```
# 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
```

Une fois les périphériques de cache ajoutés, ils se remplissent progressivement de contenu provenant de la mémoire principale. En fonction de la taille du périphérique de cache, le remplissage peut prendre plus d'une heure. La capacité et les lectures sont contrôlables à l'aide de la commande `zpool iostat` comme indiqué ci-dessous :

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

Une fois le pool créé, vous pouvez y ajouter des périphériques de cache ou les en supprimer.

Tenez compte des points suivants lorsque vous envisagez de créer un pool de stockage ZFS avec des périphériques de cache :

- L'utilisation de périphériques de cache constitue la meilleure amélioration de performances pour les charges de travail de lecture aléatoire constituées principalement de contenu statique.
- La capacité et les lectures sont contrôlables à l'aide de la commande `zpool iostat`.
- Lors de la création du pool, vous pouvez ajouter un ou plusieurs caches. Ils peuvent également être ajoutés ou supprimés après la création du pool. Pour plus d'informations, reportez-vous à l'Exemple 3–4.
- Les périphériques de cache ne peuvent pas être mis en miroir ou faire partie d'une configuration RAID-Z.
- Si une erreur de lecture est détectée sur un périphérique de cache, cette E/S de lecture est à nouveau exécutée sur le périphérique de pool de stockage d'origine, qui peut faire partie d'une configuration RAID-Z ou en miroir. Le contenu des périphériques de cache est considéré comme volatile, comme les autres caches système.

Précautions pour la création de pools de stockage

Tenez compte des mises en garde suivantes lors de la création et de la gestion de pools de stockage ZFS.

- Ne repartitionnez ou ne réétiquetez pas des disques qui font partie d'un pool de stockage existant. Si vous tentez de repartitionner ou de réétiqueter un disque de pool root, vous devrez peut-être réinstaller le système d'exploitation.
- Ne créez pas de pool de stockage contenant des composants d'un autre pool de stockage, tels que des fichiers ou des volumes. Des interblocages peuvent se produire dans cette configuration non prise en charge.
- Un pool créé avec une tranche unique ou un disque unique n'a aucune redondance et risque de perdre des données. Un pool créé avec plusieurs tranches mais sans redondance risque également de perdre des données. Un pool créé avec plusieurs tranches réparties sur plusieurs disques est plus difficile à gérer qu'un pool créé avec des disques entiers.
- Un pool créé sans redondance ZFS (RAIDZ ou miroir) peut uniquement signaler les incohérences de données. Il ne peut pas réparer les incohérences de données.
- Bien qu'un pool créé avec redondance ZFS permette de réduire le temps d'inactivité dû à des pannes matérielles, il n'est pas à l'abri de pannes matérielles, de pannes de courant ou de déconnexions de câbles. Veillez à sauvegarder régulièrement vos données. Il est important d'effectuer des sauvegardes de routine des données de pools si le matériel n'est pas de niveau professionnel.
- Un pool ne peut pas être partagé par différents systèmes. ZFS n'est pas un système de fichiers de cluster.

Affichage des informations d'un périphérique virtuel de pool de stockage

Chaque pool de stockage contient un ou plusieurs périphériques virtuels. Un *périphérique virtuel* est une représentation interne du pool de stockage qui décrit la disposition du stockage physique et les caractéristiques par défaut du pool de stockage. Ainsi, un périphérique virtuel représente les périphériques de disque ou les fichiers utilisés pour créer le pool de stockage. Un pool peut contenir un nombre quelconque de périphériques virtuels dans le niveau supérieur de la configuration. Ces périphériques sont appelés *top-level vdev*.

Si le périphérique virtuel de niveau supérieur contient deux ou plusieurs périphériques physiques, la configuration assure la redondance des données en tant que périphériques virtuels RAID-Z ou miroir. Ces périphériques virtuels se composent de disques, de tranches de disques ou de fichiers. Un disque de rechange (spare) est un périphérique virtuel spécial qui effectue le suivi des disques hot spare disponibles d'un pool.

L'exemple suivant illustre la création d'un pool composé de deux périphériques virtuels de niveau supérieur, chacun étant un miroir de deux disques :

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

L'exemple suivant illustre la création d'un pool composé d'un périphérique virtuel de niveau supérieur comportant quatre disques :

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

Vous pouvez ajouter un autre périphérique virtuel de niveau supérieur à ce pool en utilisant la commande `zpool add`. Par exemple :

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

Les disques, tranches de disque ou fichiers utilisés dans des pools non redondants fonctionnent en tant que périphériques virtuels de niveau supérieur. Les pools de stockage contiennent en règle générale plusieurs périphériques virtuels de niveau supérieur. ZFS entrelace automatiquement les données entre l'ensemble des périphériques virtuels de niveau supérieur dans un pool.

Les périphériques virtuels et les périphériques physiques contenus dans un pool de stockage ZFS s'affichent avec la commande `zpool status`. Par exemple :

```
# 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
```

Gestion d'erreurs de création de pools de stockage ZFS

Les erreurs de création de pool peuvent se produire pour de nombreuses raisons. Certaines raisons sont évidentes, par exemple lorsqu'un périphérique spécifié n'existe pas, mais d'autres le sont moins.

Détection des périphériques utilisés

Avant de formater un périphérique, ZFS vérifie que le disque n'est pas utilisé par ZFS ou une autre partie du système d'exploitation. Si le disque est en cours d'utilisation, les erreurs suivantes peuvent se produire :

```
# 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).
```

Certaines erreurs peuvent être ignorées à l'aide de l'option `-f`, mais pas toutes. Les conditions suivantes ne peuvent pas à être ignorées via l'option `-f` et doivent être corrigées manuellement :

Système de fichiers monté — Le disque ou une de ses tranches contient un système de fichiers actuellement monté. La commande `umount` permet de corriger cette erreur.

Système de fichiers dans /etc/vfstab — Le disque contient un système de fichiers répertorié dans le fichier `/etc/vfstab`, mais le système de fichiers n'est pas monté. Pour corriger cette erreur, supprimez ou commentez la ligne dans le fichier `/etc/vfstab`.

Périphérique de vidage dédié — Le disque est utilisé en tant que périphérique de vidage dédié pour le système. La commande `dumpadm` permet de corriger cette erreur.

Elément d'un pool ZFS — Le disque ou fichier fait partie d'un pool de stockage ZFS. Pour corriger cette erreur, utilisez la commande `zpool destroy` afin de détruire l'autre pool s'il est

obsolète. Utilisez sinon la commande `zpool detach` pour déconnecter le disque de l'autre pool. Vous pouvez déconnecter un disque que s'il est connecté à un pool de stockage mis en miroir.

Les vérifications en cours d'utilisation suivantes constituent des avertissements. Pour les ignorer, appliquez l'option `-f` afin de créer le pool :

Contient un système de fichiers — Le disque contient un système de fichiers connu bien qu'il ne soir pas monté et n'apparaisse pas comme étant en cours d'utilisation.

Elément d'un volume — Le disque fait partie d'un volume Solaris Volume Manager.

Live upgrade — Le disque est en cours d'utilisation en tant qu'environnement d'initialisation de remplacement pour Oracle Solaris Live Upgrade.

Elément d'un pool ZFS exporté — Le disque fait partie d'un pool de stockage exporté ou supprimé manuellement d'un système. Dans le deuxième cas, le pool est signalé comme étant `potentiellement actif`, dans la mesure où il peut s'agir d'un disque connecté au réseau en cours d'utilisation par un autre système. Faites attention lorsque vous ignorez un pool potentiellement activé.

L'exemple suivant illustre l'utilisation de l'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
```

Si possible, corrigez les erreurs au lieu d'utiliser l'option `-f` pour les ignorer.

Niveaux de réplication incohérents

Il est déconseillé de créer des pools avec des périphériques virtuels de niveau de réplication différents. La commande `zpool` tente de vous empêcher de créer par inadvertance un pool comprenant des niveaux de redondance différents. Si vous tentez de créer un pool avec un telle configuration, les erreurs suivantes s'affichent :

```
# 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
```

Vous pouvez ignorer ces erreurs avec l'option `-f`. Toutefois, cette pratique est déconseillée. La commande affiche également un avertissement relatif à la création d'un pool RAID-Z ou mis en miroir à l'aide de périphériques de tailles différentes. Même si cette configuration est autorisée, les niveaux de redondance sont incohérents. Par conséquent, l'espace disque du périphérique de plus grande taille n'est pas utilisé. Vous devez spécifier l'option `-f` pour ignorer l'avertissement.

Réalisation d'un test à la création d'un pool de stockage

Les tentatives de création d'un pool peuvent échouer soudainement de plusieurs façons ; vous pouvez formater les disques, mais cela peut avoir des conséquences négatives. C'est pourquoi la commande `zpool create` dispose d'une option supplémentaire, à savoir l'option `-n`, qui simule la création du pool sans écrire les données sur le périphérique. Cette option de *test* vérifie le périphérique en cours d'utilisation et valide le niveau de réplication, puis répertorie les erreurs survenues au cours du processus. Si aucune erreur n'est détectée, la sortie est similaire à la suivante :

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

        tank
          mirror
            c1t0d0
            c1t1d0
```

Certaines erreurs sont impossibles à détecter sans création effective du pool. L'exemple le plus courant consiste à spécifier le même périphérique deux fois dans la même configuration. Cette erreur ne peut pas être détectée de façon fiable sans l'enregistrement effectif des données. Par conséquent, la commande `zpool create -n` peut indiquer que l'opération a réussi sans pour autant parvenir à créer le pool, lors de son exécution sans cette option.

Point de montage par défaut pour les pools de stockage

Lors de la création d'un pool, le point de montage par défaut du système de fichiers de niveau supérieur est */pool-name* . Le répertoire doit être inexistant ou vide. Le répertoire est créé automatiquement s'il n'existe pas. Si le répertoire est vide, le système de fichiers root est monté sur le répertoire existant. Pour créer un pool avec un point de montage par défaut différent, utilisez l'option `-m` de la commande `zpool create` : Par exemple :

```
# 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
```

Cette commande crée le pool `home` et le système de fichiers `home` avec le point de montage `/export/zfs`.

Pour de plus amples informations sur les points de montage, reportez-vous à la section "Gestion des points de montage ZFS" à la page 206.

Destruction de pools de stockage ZFS

La commande `zpool destroy` permet de détruire les pools. Cette commande détruit le pool même s'il contient des jeux de données montés.

```
# zpool destroy tank
```

Attention – Faites très attention lorsque vous détruisez un pool. Assurez-vous de détruire le pool souhaité et de toujours disposer de copies de vos données. En cas de destruction accidentelle d'un pool, vous pouvez tenter de le récupérer. Pour obtenir des informations supplémentaires, reportez-vous à la section "Récupération de pools de stockage ZFS détruits" à la page 107.

Si vous détruisez un pool à l'aide de la commande `zpool destroy`, le pool reste disponible pour l'importation, comme décrit dans la section "Récupération de pools de stockage ZFS détruits" à la page 107. Cela signifie que des données confidentielles peuvent subsister sur les disques qui faisaient partie du pool. Si vous souhaitez détruire les données placées sur les disques du pool détruit, vous devez utiliser une fonctionnalité telle que l'option `analyze->purge` de l'utilitaire `format` sur tous les disques du pool détruit.

Destruction d'un pool avec des périphériques disponibles

La destruction d'un pool requiert l'écriture des données sur le disque pour indiquer que le pool n'est désormais plus valide. Ces informations d'état évitent que les périphériques ne s'affichent en tant que pool potentiel lorsque vous effectuez une importation. La destruction du pool est tout de même possible si un ou plusieurs périphériques ne sont pas disponibles. Cependant, les informations d'état requises ne sont pas écrites sur ces périphériques indisponibles.

Ces périphériques, lorsqu'ils sont correctement réparés, sont signalés comme *potentiellement actifs*, lors de la création d'un pool. Lorsque vous recherchez des pools à importer, ils s'affichent en tant que périphériques valides. Si un pool a tant de périphériques `UNAVAIL` qu'il est lui-même `UNAVAIL` (en d'autres termes, un périphérique virtuel de niveau supérieur est `UNAVAIL`), la commande émet un avertissement et ne peut pas s'exécuter sans l'option `-f`. Cette option est requise car l'ouverture du pool est impossible et il est impossible de savoir si des données y sont stockées. Par exemple :

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

Pour de plus amples informations sur les pools et la maintenance des périphériques, reportez-vous à la section "Détermination de l'état de maintenance des pools de stockage ZFS" à la page 94.

Pour de plus amples informations sur l'importation de pools, reportez-vous à la section "Importation de pools de stockage ZFS" à la page 103.

Gestion de périphériques dans un pool de stockage ZFS

Vous trouverez la plupart des informations de base concernant les périphériques dans la section "Composants d'un pool de stockage ZFS" à la page 45. Après la création d'un pool, vous pouvez effectuer plusieurs tâches de gestion des périphériques physiques au sein du pool.

- "Ajout de périphériques à un pool de stockage" à la page 64
- "Connexion et séparation de périphériques dans un pool de stockage " à la page 69
- "Création d'un pool par scission d'un pool de stockage ZFS mis en miroir" à la page 71
- "Mise en ligne et mise hors ligne de périphériques dans un pool de stockage" à la page 74
- "Effacement des erreurs de périphérique de pool de stockage" à la page 77
- "Remplacement de périphériques dans un pool de stockage" à la page 77
- "Désignation des disques hot spare dans le pool de stockage" à la page 80

Ajout de périphériques à un pool de stockage

Vous pouvez ajouter de l'espace disque à un pool de façon dynamique, en ajoutant un périphérique virtuel de niveau supérieur. Cet espace disque est disponible immédiatement pour l'ensemble des jeux de données du pool. Pour ajouter un périphérique virtuel à un pool, utilisez la commande `zpool add`. Par exemple :

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

Le format de spécification des périphériques virtuels est le même que pour la commande `zpool create`. Une vérification des périphériques est effectuée afin de déterminer s'ils sont en cours d'utilisation et la commande ne peut pas modifier le niveau de redondance sans l'option `-f`. La commande prend également en charge l'option `-n`, ce qui permet d'effectuer un test. Par exemple :

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

```
        mirror
            c1t0d0
            c1t1d0
        mirror
            c2t1d0
            c2t2d0
        mirror
            c3t1d0
            c3t2d0
```

Cette syntaxe de commande ajouterait les périphériques en miroir `c3t1d0` et `c3t2d0` à la configuration existante du pool `zeepool`.

Pour plus d'informations sur la validation des périphériques virtuels, reportez-vous à la section "Détection des périphériques utilisés" à la page 60.

EXEMPLE 3–1 Ajout de disques à une configuration ZFS mise en miroir

Dans l'exemple suivant, un autre miroir est ajouté à une configuration ZFS mise en miroir existante.

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

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

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

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

EXEMPLE 3–1 Ajout de disques à une configuration ZFS mise en miroir *(Suite)*

```
errors: No known data errors
```

EXEMPLE 3–2 Ajout de disques à une configuration RAID-Z

De la même façon, vous pouvez ajouter des disques supplémentaires à une configuration RAID-Z. L'exemple suivant illustre la conversion d'un pool de stockage avec un périphérique RAID–Z composé de trois disques en pool de stockage avec deux périphériques RAID-Z composés de trois disques chacun.

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

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

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

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

errors: No known data errors
```

EXEMPLE 3–3 Ajout et suppression d'un périphérique de journalisation mis en miroir

L'exemple suivant indique comment ajouter un périphérique de journalisation mis en miroir dans un pool de stockage mis en miroir.

```
# zpool status newpool
  pool: newpool
 state: ONLINE
```

EXEMPLE 3–3 Ajout et suppression d'un périphérique de journalisation mis en miroir *(Suite)*

```
 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
```

Vous pouvez connecter un périphérique de journalisation à un périphérique de journalisation existant afin de créer un périphérique mis en miroir. Cette opération est similaire à la connexion d'un périphérique à un pool de stockage qui n'est pas mis en miroir.

Vous pouvez supprimer les périphériques de journalisation en utilisant la commande `zpool remove`. Le périphérique de journalisation mis en miroir dans l'exemple précédent peut être supprimé en spécifiant l'argument `miroir-1`. Par exemple :

```
# 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
```

EXEMPLE 3–3 Ajout et suppression d'un périphérique de journalisation mis en miroir *(Suite)*

Si votre configuration de pool contient un seul périphérique de journalisation, supprimez-le en saisissant le nom du périphérique. Par exemple :

```
# 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
```

EXEMPLE 3–4 Ajout et suppression des périphériques de cache

Vous pouvez ajouter des périphériques de cache à votre pool de stockage ZFS et les supprimer s'ils ne sont plus nécessaires.

Utilisez la commande `zpool add` pour ajouter des périphériques de cache. Par exemple :

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

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

errors: No known data errors
```

Les périphériques de cache ne peuvent pas être mis en miroir ou faire partie d'une configuration RAID-Z.

Utilisez la commande `zpool remove` pour supprimer des périphériques de cache. Par exemple :

EXEMPLE 3–4 Ajout et suppression des périphériques de cache *(Suite)*

```
# 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
```

Actuellement, la commande `zpool remove` prend uniquement en charge la suppression des disques hot spare, des périphériques de journalisation et des périphériques de cache. Les périphériques faisant partie de la configuration de pool mis en miroir principale peuvent être supprimés à l'aide de la commande `zpool detach`. Les périphériques non redondants et RAID-Z ne peuvent pas être supprimés d'un pool.

Pour plus d'informations sur l'utilisation des périphériques de cache dans un pool de stockage ZFS, reportez-vous à la section “Création d'un pool de stockage ZFS avec des périphériques de cache” à la page 57.

Connexion et séparation de périphériques dans un pool de stockage

Outre la commande `zpool add`, vous pouvez utiliser la commande `zpool attach` pour ajouter un périphérique à un périphérique existant, en miroir ou non.

Si vous connectez un disque pour créer un pool root mis en miroir, reportez-vous à la section “Création d'un pool root ZFS mis en miroir (post-installation)” à la page 122.

Si vous remplacez un disque dans un pool root ZFS, reportez-vous à la section “Restauration du pool root ZFS ou des instantanés du pool root” à la page 170.

EXEMPLE 3–5 Conversion d'un pool de stockage bidirectionnel mis en miroir en un pool de stockage tridirectionnel mis en miroir

Dans cet exemple, `zeepool` est un miroir bidirectionnel. Il est converti en un miroir tridirectionnel via la connexion de `c2t1d0`, le nouveau périphérique, au périphérique existant, `c1t1d0`.

EXEMPLE 3–5 Conversion d'un pool de stockage bidirectionnel mis en miroir en un pool de stockage tridirectionnel mis en miroir *(Suite)*

```
# 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
```

Si le périphérique existant fait partie d'un miroir tridirectionnel, la connexion d'un nouveau périphérique crée un miroir quadridirectionnel, et ainsi de suite. Dans tous les cas, la réargenture du nouveau périphérique commence immédiatement.

EXEMPLE 3–6 Conversion d'un pool de stockage ZFS non redondant en pool de stockage ZFS en miroir

En outre, vous pouvez convertir un pool de stockage non redondant en pool de stockage redondant à l'aide de la commande `zpool attach`. Par exemple :

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

errors: No known data errors
# zpool attach tank c0t1d0 c1t1d0
# zpool status tank
  pool: tank
```

EXEMPLE 3–6 Conversion d'un pool de stockage ZFS non redondant en pool de stockage ZFS en miroir *(Suite)*

```
 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
```

Vous pouvez utiliser la commande `zpool detach` pour séparer un périphérique d'un pool de stockage mis en miroir. Par exemple :

```
# zpool detach zeepool c2t1d0
```

Cependant, en l'absence de répliques de données valides, cette opération échoue. Par exemple :

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

Création d'un pool par scission d'un pool de stockage ZFS mis en miroir

Un pool de stockage ZFS mis en miroir peut être rapidement cloné en tant que pool de sauvegarde à l'aide de la commande `zpool split`. Vous pouvez utiliser cette fonctionnalité pour scinder un pool root mis en miroir. Toutefois, le pool qui a été séparé n'est pas amorçable tant que vous n'avez pas suivi des étapes supplémentaires.

Vous pouvez utiliser la commande `zpool split` pour déconnecter un ou plusieurs disques à partir d'un pool de stockage ZFS mis en miroir afin de créer un pool de stockage avec l'un des disques déconnectés. Le nouveau pool contiendra les mêmes données que le pool de stockage ZFS d'origine mis en miroir.

Par défaut, une opération `zpool split` sur un pool mis en miroir déconnecte le dernier disque du nouveau pool. Une fois l'opération de scission terminée, importez le nouveau pool. Par exemple :

```
# 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
```

Vous pouvez identifier le disque à utiliser par le nouveau pool en le définissant avec la commande `zpool split`. Par exemple :

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

Avant l'opération de scission, les données de la mémoire sont vidées vers les disques mis en miroir. Une fois les données vidées, le disque est déconnecté du pool et reçoit un nouveau GUID de pool. Un pool GUID est généré pour que le pool puisse être importé sur le même système que celui sur lequel il a été scindé.

Si le pool à scinder possède des points de montage de système de fichiers autres que celui par défaut et si le nouveau pool est créé sur le même système, utilisez l'option `zpool split -R` pour identifier un autre répertoire root pour le nouveau pool afin d'éviter tout conflit entre les points de montage existants. Par exemple :

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

Si vous n'utilisez pas l'option `zpool split -R` et si des points de montage entrent en conflit lorsque vous tentez d'importer le nouveau pool, importez celui-ci à l'aide de l'option `-R`. Si le

nouveau pool est créé sur un autre système, il n'est pas nécessaire de spécifier un autre répertoire root, sauf en cas de conflits entre les points de montage.

Avant d'utiliser la fonctionnalité `zpool split`, veuillez prendre en compte les points suivants :

- Cette fonction n'est pas disponible dans une configuration RAID-Z ou un pool non redondant composé de plusieurs disques.
- Avant de tenter une opération `zpool split`, les opérations des données et des applications doivent être suspendues.
- Vous ne pouvez pas scinder un pool si une réargenture est en cours.
- Lorsqu'un pool mis en miroir est composé de deux à trois disques et que le dernier disque du pool d'origine est utilisé pour le nouveau pool créé, la meilleure solution consiste à scinder le pool mis en miroir. Vous pouvez ensuite utiliser la commande `zpool attach` pour recréer votre pool de stockage d'origine mis en miroir ou convertir votre nouveau pool dans un pool de stockage mis en miroir. Aucune méthode ne permet de créer un *nouveau* pool mis en miroir à partir d'un pool mis en miroir *existant* en une opération `zpool split`, car le nouveau pool (séparé) n'est pas redondant.
- Si le pool existant est un miroir tridirectionnel, le nouveau pool contiendra un disque après l'opération de scission. Si le pool existant est un miroir bidirectionnel composé de deux disques, cela donne deux pools non redondants composés de deux disques. Vous devez connecter deux disques supplémentaires pour convertir les pools non redondants en pools mis en miroir.
- Pour conserver vos données redondantes lors d'une scission, scindez un pool de stockage mis en miroir composé de trois disques pour que le pool d'origine soit composé de deux disques mis en miroir après la scission.
- Confirmez que votre matériel est configuré correctement avant de séparer un pool mis en miroir. Pour plus d'informations sur la confirmation des paramètres de purge du cache de votre matériel, reportez-vous à la section “Pratiques recommandées générales” à la page 327.

EXEMPLE 3–7 Scission d'un pool ZFS mis en miroir

Dans l'exemple suivant, un pool de stockage mis en miroir, nommé `mothership` avec trois disques, est séparé. Les deux pools obtenus sont le pool mis en miroir `mothership`, avec deux disques, et le nouveau pool `luna` , avec un disque. Chaque pool contient les mêmes données.

Le pool `luna` peut être importé dans un autre système à des fins de sauvegarde. Une fois la sauvegarde terminée, le pool `luna` peut être détruit et rattaché à `mothership`. Le processus peut ensuite être répété.

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

EXEMPLE 3–7 Scission d'un pool ZFS mis en miroir *(Suite)*

```
        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
```

Mise en ligne et mise hors ligne de périphériques dans un pool de stockage

ZFS permet la mise en ligne ou hors ligne de périphériques. Lorsque le matériel n'est pas fiable ou fonctionne mal, ZFS continue de lire ou d'écrire les données dans le périphérique en partant du principe que le problème est temporaire. Dans le cas contraire, vous pouvez indiquer à ZFS d'ignorer le périphérique en le mettant hors ligne. Le système de fichiers ZFS n'envoie aucune demande à un périphérique déconnecté.

Remarque – Il est inutile de mettre les périphériques hors ligne pour les remplacer.

Mise hors ligne d'un périphérique

La commande `zpool offline` permet de mettre un périphérique hors ligne. Vous pouvez spécifier le périphérique via son chemin ou via son nom abrégé s'il s'agit d'un disque. Par exemple :

```
# zpool offline tank c0t5000C500335F95E3d0
```

Lors de la déconnexion d'un périphérique, veuillez prendre en compte les points suivants :

- Vous ne pouvez pas mettre un périphérique hors ligne au point où il devient `UNAVAIL`. Par exemple, vous ne pouvez pas mettre hors ligne deux périphériques d'une configuration `raid-z1`, ni ne pouvez mettre hors ligne un périphérique virtuel de niveau supérieur.

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

- Par défaut, l'état `OFFLINE` est persistant. Le périphérique reste hors ligne lors de la réinitialisation du système.

  Pour mettre un périphérique hors ligne temporairement, utilisez l'option `-t` de la commande `zpool offline`. Par exemple :

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

  En cas de réinitialisation du système, ce périphérique revient automatiquement à l'état `ONLINE`.

- Lorsqu'un périphérique est mis hors ligne, il n'est pas séparé du pool de stockage. En cas de tentative d'utilisation du périphérique hors ligne dans un autre pool, même en cas de destruction du pool d'origine, un message similaire au suivant s'affiche :

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

  Si vous souhaitez utiliser le périphérique hors ligne dans un autre pool de stockage après destruction du pool de stockage d'origine, remettez le périphérique en ligne puis détruisez le pool de stockage d'origine.

  Une autre mode d'utilisation d'un périphérique provenant d'un autre pool de stockage si vous souhaitez conserver le pool de stockage d'origine consiste à remplacer le périphérique existant dans le pool de stockage d'origine par un autre périphérique similaire. Pour obtenir des informations sur le remplacement de périphériques, reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 77.

Les périphériques mis hors ligne s'affichent dans l'état `OFFLINE` en cas de requête de l'état de pool. Pour obtenir des informations sur les requêtes d'état de pool, reportez-vous à la section “Requête d'état de pool de stockage ZFS” à la page 88.

Pour de plus amples informations sur la maintenance des périphériques, reportez-vous à la section “Détermination de l'état de maintenance des pools de stockage ZFS” à la page 94.

Mise en ligne d'un périphérique

Lorsqu'un périphérique est mis hors ligne, il peut être restauré grâce à la commande `zpool online`. Par exemple :

```
# zpool online tank c0t5000C500335F95E3d0
```

Lorsqu'un périphérique est mis en ligne, toute donnée écrite dans le pool est resynchronisée sur le périphérique nouvellement disponible. Notez que vous ne pouvez pas mettre en ligne un périphérique pour remplacer un disque. Si vous mettez un périphérique hors ligne, le remplacez, puis tentez de le mettre en ligne, son état reste `UNAVAIL` .

Si vous tentez de mettre un périphérique `UNAVAIL` en ligne, un message similaire au suivant s'affiche :

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

Vous pouvez également afficher les messages de disques erronés dans la console ou les messages enregistrés dans le fichier `/var/adm/messages`. Par exemple :

```
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.
```

Pour obtenir des informations sur le remplacement d'un périphérique défaillant, reportez-vous à la section "Résolution d'un périphérique manquant ou supprimé" à la page 301.

Vous pouvez utiliser la commande `zpool online -e` pour développer la taille du pool si un plus grand disque a été associé au pool ou si un petit disque a été remplacé par un plus grand disque. Par défaut, un disque ajouté à un pool n'est pas étendu à sa taille maximale, à moins que la propriété `autoexpand` du pool ne soit activée. Vous pouvez étendre automatiquement le pool en utilisant la commande `zpool online -e`, même si le disque de remplacement est déjà en ligne ou s'il est actuellement hors ligne. Par exemple :

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

Effacement des erreurs de périphérique de pool de stockage

Si un périphérique est mis hors ligne en raison d'une défaillance qui entraîne l'affichage d'erreurs dans la sortie `zpool status`, la commande `zpool clear` permet d'effacer les nombres d'erreurs.

Si elle est spécifiée sans argument, cette commande efface toutes les erreurs de périphérique dans le pool. Par exemple :

```
# zpool clear tank
```

Si un ou plusieurs périphériques sont spécifiés, cette commande n'efface que les erreurs associées aux périphériques spécifiés. Par exemple :

```
# zpool clear tank c0t5000C500335F95E3d0
```

Pour de plus amples informations sur l'effacement d'erreurs de `zpool`, reportez-vous à la section "Suppression des erreurs de périphérique transitoires" à la page 306.

Remplacement de périphériques dans un pool de stockage

Vous pouvez remplacer un périphérique dans un pool de stockage à l'aide de la commande `zpool replace`.

Pour remplacer physiquement un périphérique par un autre, en conservant le même emplacement dans le pool redondant, il vous suffit alors d'identifier le périphérique remplacé. Sur certains matériels, ZFS reconnaît que le périphérique est un disque différent au même emplacement. Par exemple, pour remplacer un disque défaillant (`c1t1d0`), supprimez-le, puis ajoutez le disque de rechange au même emplacement en respectant la syntaxe suivante :

```
# zpool replace tank c1t1d0
```

Si vous remplacez un périphérique dans un pool de stockage par un disque dans un autre emplacement physique, vous devez spécifier les deux périphériques. Par exemple :

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

Si vous remplacez un disque dans un pool root ZFS, reportez-vous à la section "Restauration du pool root ZFS ou des instantanés du pool root" à la page 170.

Voici les étapes de base pour remplacer un disque :

1. Le cas échéant, mettez le disque hors ligne à l'aide de la commande `zpool offline`.
2. Enlevez le disque à remplacer.
3. Insérez le disque de remplacement.
4. Consultez la sortie de `format` pour déterminer si le disque de remplacement est visible.

   Vérifiez également si l'ID du périphérique a changé. Si le disque de remplacement a un nom universel, alors l'ID de périphérique du disque défaillant a changé.
5. Indiquez au système ZFS que le disque a été remplacé. Par exemple :

   ```
   # zpool replace tank c1t1d0
   ```

   Si le disque de remplacement a un ID de périphérique différent, incluez ce dernier.

   ```
   # zpool replace tank c0t5000C500335FC3E7d0 c0t5000C500335BA8C3d0
   ```
6. Remettez le disque en ligne à l'aide de la commande `zpool online`, si nécessaire.
7. Informez FMA du remplacement du périphérique.

   Dans la sortie `fmadm faulty`, identifiez la chaîne `zfs://pool=name/vdev=guid` dans la section `Affects:` et attribuez-la comme argument à la commande `fmadm repaired`.

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

Sur certains systèmes avec des disques SATA, vous devez annuler la configuration d'un disque avant de pouvoir mettre hors ligne. Si vous remplacez un disque dans le même emplacement sur ce système, vous pouvez exécuter la commande `zpool replace` comme décrit dans le premier exemple de cette section.

Pour consulter un exemple de remplacement d'un disque SATA, reportez-vous à l'Exemple 10–1.

Lorsque vous remplacez des périphériques dans un pool de stockage ZFS, veuillez prendre en compte les points suivants :

- Si vous définissez la propriété de pool `autoreplace` sur `on`, tout nouveau périphérique détecté au même emplacement physique qu'un périphérique appartenant précédemment au pool est automatiquement formaté et remplacé. Lorsque cette propriété est activée, vous n'êtes pas obligé d'utiliser la commande `zpool replace`. Cette fonction n'est pas disponible sur tous les types de matériel.
- L'état de pool de stockage `REMOVED` est fourni en cas de retrait physique du périphérique ou d'un disque hot spare alors que le système est en cours d'exécution. Si un disque hot spare est disponible, il remplace le périphérique retiré.
- Si un périphérique est retiré, puis réinséré, il est mis en ligne. Si un disque hot spare est activé lors de la réinsertion du périphérique, le disque hot spare est retiré une fois l'opération en ligne terminée.

- La détection automatique du retrait ou de l'insertion de périphériques dépend du matériel utilisé. Il est possible qu'elle ne soit pas prise en charge sur certaines plates-formes. Par exemple, les périphériques USB sont configurés automatiquement après insertion. Il peut être toutefois nécessaire d'utiliser la commande `cfgadm -c configure` pour configurer un lecteur SATA.
- Les disques hot spare sont consultés régulièrement afin de vérifier qu'ils sont en ligne et disponibles.
- La taille du périphérique de remplacement doit être égale ou supérieure au disque le plus petit d'une configuration RAID-Z ou mise en miroir.
- Lorsqu'un périphérique de remplacement dont la taille est supérieure à la taille du périphérique qu'il remplace est ajouté à un pool, ce dernier n'est pas automatiquement étendu à sa taille maximale. La valeur de propriété `autoexpand` du pool détermine si le pool est développé lorsqu'un plus grand disque est ajouté au pool. Par défaut, la propriété `autoexpand` est désactivée. Vous pouvez activer cette propriété pour augmenter la taille du pool avant ou après avoir ajouté le plus grand disque au pool.

  Dans l'exemple suivant, deux disques de 16 Go d'un pool mis en miroir sont remplacés par deux disques de 72 Go. Assurez-vous qu'une réargenture complète a été effectuée sur le premier périphérique avant de tenter le remplacement du deuxième périphérique. La propriété `autoexpand` est activée après les remplacements de disque pour étendre le disque à sa taille maximale.

```
# 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  -
```

- Le remplacement d'un grand nombre de disques dans un pool volumineux prend du temps, en raison de la réargenture des données sur les nouveaux disques. En outre, il peut s'avérer utile d'exécuter la commande `zpool scrub` entre chaque remplacement de disque afin de garantir le fonctionnement des périphériques de remplacement et l'exactitude des données écrites.
- Si un disque défectueux a été remplacé automatiquement par un disque hot spare, il se peut que vous deviez déconnecter le disque hot spare une fois le disque défectueux remplacé. Vous pouvez utiliser la commande `zpool detach` pour déconnecter le disque hot spare d'un pool RAID-Z ou mis en miroir. Pour plus d'informations sur la déconnexion d'un disque hot spare, reportez-vous à la section "Activation et désactivation de disques hot spare dans le pool de stockage" à la page 82.

Pour plus d'informations sur le remplacement de périphériques, reportez-vous aux sections "Résolution d'un périphérique manquant ou supprimé" à la page 301 et "Remplacement ou réparation d'un périphérique endommagé " à la page 304.

Désignation des disques hot spare dans le pool de stockage

La fonction de disques hot spare permet d'identifier les disques utilisables pour remplacer un périphérique défaillant dans un pool de stockage. Un périphérique désigné en tant que *disque hot spare* n'est pas actif dans un pool, mais en cas d'échec d'un périphérique actif du pool, le disque hot spare le remplace automatiquement

Pour désigner des périphériques en tant que disques hot spare, vous avez le choix entre les méthodes suivantes :

- lors de la création du pool à l'aide de la commande `zpool create` ;
- après la création du pool à l'aide de la commande `zpool create`.

L'exemple suivant explique comment désigner des périphériques en tant que disques hot spare lorsque le pool est créé :

```
# 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
```

L'exemple suivant explique comment désigner des disques hot spare en les ajoutant à un pool après la création du pool :

```
# 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
```

Vous pouvez supprimer les disques hot spare d'un pool de stockage à l'aide de la commande `zpool remove`. Par exemple :

```
# 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
```

Vous ne pouvez pas supprimer un disque hot spare si ce dernier est actuellement utilisé par un pool de stockage.

Lorsque vous utilisez des disques hot spare ZFS, veuillez prendre en compte les points suivants :

- Actuellement, la commande `zpool remove` ne peut être utilisée que pour la suppression de disques hot spare, de périphériques de journalisation et de périphériques de cache.
- Pour ajouter un disque en tant que disque hot spare, la taille du disque hot spare doit être égale ou supérieure à la taille du plus grand disque du pool. L'ajout d'un disque de rechange plus petit dans le pool est autorisé. Toutefois, lorsque le plus petit disque de rechange est activé, automatiquement ou via la commande `zpool replace`, l'opération échoue et une erreur du type suivant s'affiche :

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

Activation et désactivation de disques hot spare dans le pool de stockage

Les disques hot spare s'activent des façons suivantes :

- Remplacement manuel : remplacez un périphérique défaillant dans un pool de stockage par un disque hot spare à l'aide de la commande `zpool replace`.
- Remplacement automatique : en cas de détection d'une défaillance, un agent FMA examine le pool pour déterminer s'il y a des disques hot spare. Dans ce cas, le périphérique défaillant est remplacé par un disque hot spare disponible.

  En cas de défaillance d'un disque hot spare en cours d'utilisation, l'agent FMA sépare le disque hot spare et annule ainsi le remplacement. L'agent tente ensuite de remplacer le périphérique par un autre disque hot spare s'il y en a un de disponible. Cette fonction est actuellement limitée par le fait que le moteur de diagnostics ZFS ne génère des défaillances qu'en cas de disparition d'un périphérique du système.

  Si vous remplacez physiquement un périphérique défaillant par un disque spare actif, vous pouvez réactiver le périphérique original en utilisant la commande `zpool detach` pour déconnecter le disque spare. Si vous définissez la propriété de pool `autoreplace` sur `on`, le disque spare est automatiquement déconnecté et retourne au pool de disques spare lorsque le nouveau périphérique est inséré et que l'opération en ligne s'achève.

Tout périphérique `UNAVAIL` est remplacé automatiquement si un disque hot spare est disponible. Par exemple :

```
# 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
```

Vous pouvez actuellement désactiver un disque hot spare en recourant à l'une des méthodes suivantes :

- Suppression du disque hot spare du pool de stockage.
- Déconnexion du disque hot spare après avoir remplacé physiquement un disque défectueux. Reportez-vous à l'Exemple 3–8.
- Remplacement temporaire ou permanent par un autre disque hot spare. Reportez-vous à l'Exemple 3–9.

EXEMPLE 3–8 Déconnexion d'un disque hot spare après le remplacement du disque défectueux

Dans cet exemple, le disque défectueux (`c0t5000C500335DC60Fd0`) est remplacé physiquement et ZFS est averti à l'aide de la commande `zpool replace`.

```
# 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
```

Si nécessaire, vous pouvez ensuite utiliser la commande `zpool detach` pour retourner le disque hot spare au pool de disques hot spare. Par exemple :

EXEMPLE 3–8 Déconnexion d'un disque hot spare après le remplacement du disque défectueux *(Suite)*

```
# zpool detach zeepool c0t5000C500335E106Bd0
```

EXEMPLE 3–9 Déconnexion d'un disque défectueux et utilisation d'un disque hot spare

Si vous souhaitez remplacer un disque défectueux par un swap temporaire ou permanent dans le disque hot spare qui le remplace actuellement, vous devez déconnecter le disque d'origine (défectueux). Si le disque défectueux finit par être remplacé, vous pouvez l'ajouter de nouveau au groupe de stockage en tant que disque hot spare. Par exemple :

```
# 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)
```

EXEMPLE 3–9 Déconnexion d'un disque défectueux et utilisation d'un disque hot spare *(Suite)*

```
# 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:

        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
```

Une fois qu'un disque est remplacé et que le remplacement est détaché, informez FMA que le disque est réparé.

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

Gestion des propriétés de pool de stockage ZFS

Vous pouvez vous servir de la commande `zpool get` pour afficher des informations sur les propriétés du pool. Par exemple :

```
# 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                  -
```

Les propriétés d'un pool de stockage peuvent être définies à l'aide de la commande `zpool set`. Par exemple :

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

Si vous tentez de définir une propriété de pool sur un pool à 100% de sa capacité, un message semblable à celui-ci s'affiche :

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

Pour plus d'informations sur la prévention des problèmes de capacité d'espace des pools, reportez-vous au Chapitre 11, “Pratiques recommandées pour Oracle Solaris ZFS”.

TABLEAU 3–1 Description des propriétés d'un pool ZFS

Nom de propriété	Type	Valeur par défaut	Description
`allocated`	Chaîne	SO	Valeur en lecture seule permettant d'identifier l'espace de stockage disponible physiquement alloué dans le pool.
`altroot`	Chaîne	`off`	Identifie un répertoire root alternatif. S'il est défini, ce répertoire est ajouté au début de tout point de montage figurant dans le pool. Cette propriété peut être utilisée lors de l'examen d'un pool inconnu si les points de montage ne sont pas de confiance ou dans un environnement d'initialisation alternatif dans lequel les chemins types sont incorrects.
`autoreplace`	Booléen	`off`	Contrôle le remplacement automatique d'un périphérique. Si la valeur `off` est définie, le remplacement du périphérique doit être initié à l'aide de la commande `zpool replace`. Si la valeur est définie sur `on`, tout nouveau périphérique se trouvant au même emplacement physique qu'un périphérique qui appartenait au pool est automatiquement formaté et remplacé. L'abréviation de la propriété est la suivante : `replace`.
`bootfs`	Booléen	SO	Identifie le système de fichiers amorçable par défaut du pool root. Cette propriété est généralement définie par les programmes d'installation.

TABLEAU 3–1 Description des propriétés d'un pool ZFS *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`cachefile`	Chaîne	SO	Contrôle l'emplacement de la mise en cache du pool. Tous les pools du cache sont importés automatiquement à l'initialisation du système. Toutefois, dans les environnements d'installation et de clustering, il peut s'avérer nécessaire de placer ces informations en cache à un autre endroit afin d'éviter l'importation automatique des pools. Vous pouvez définir cette propriété pour mettre en cache les informations de configuration du pool dans un autre emplacement. Ces informations peuvent être importées ultérieurement à l'aide de la commande `zpool import -c`. Pour la plupart des configurations ZFS, cette propriété n'est pas utilisée.
`capacity`	Valeur numérique	SO	Valeur en lecture seule identifiant le pourcentage d'espace utilisé du pool. L'abréviation de la propriété est `cap`.
`delegation`	Booléen	`on`	Contrôle si un utilisateur non privilégié peut bénéficier des autorisations d'accès définies pour un système de fichiers. Pour plus d'informations, reportez-vous au Chapitre 8, “Administration déléguée de ZFS dans Oracle Solaris”.
`failmode`	Chaîne	`wait`	Contrôle le comportement du système en cas de panne grave d'un pool. Cette condition résulte habituellement d'une perte de connectivité aux périphériques de stockage sous-jacents ou d'une panne de tous les périphériques au sein du pool. Le comportement d'un événement de ce type est déterminé par l'une des valeurs suivantes : ■ `wait` : bloque toutes les demandes d'E/S vers le pool jusqu'au rétablissement de la connectivité et jusqu'à l'effacement des erreurs à l'aide de la commande `zpool clear`. Dans cet état, les opérations d'E/S du pool sont bloquées mais les opérations de lecture peuvent aboutir. Un pool renvoie l'état `wait` jusqu'à ce que le problème du périphérique soit résolu. ■ `continue` : renvoie une erreur EIO à toute nouvelle demande d'E/S d'écriture, mais autorise les lectures de tout autre périphérique fonctionnel. Toute demande d'écriture devant encore être validée sur disque est bloquée. Une fois le périphérique reconnecté ou remplacé, les erreurs doivent être effacées à l'aide de la commande `zpool clear`. ■ `panic` : affiche un message sur la console et génère un vidage sur incident du système.

TABLEAU 3–1 Description des propriétés d'un pool ZFS *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`free`	Chaîne	SO	Valeur en lecture seule identifiant le nombre de blocs non alloués au sein du pool.
`guid`	Chaîne	SO	Propriété en lecture seule identifiant l'identificateur unique du pool.
`health`	Chaîne	SO	Propriété en lecture seule indiquant l'état actuel du pool ; les valeurs possibles sont : ONLINE, DEGRADED, SUSPENDED, REMOVED ou UNAVAIL.
`listshares`	Chaîne	off	Contrôle si des informations de partage dans ce pool sont affichées avec la commande `zfs list`. La valeur par défaut est `off`.
`listsnapshots`	Chaîne	`on`	Détermine si les informations sur les instantanés associées à ce groupe s'affichent avec la commande `zfs list`. Si cette propriété est désactivée, les informations sur les instantanés peuvent être affichées à l'aide de la commande `zfs list -t snapshot`.
`size`	Valeur numérique	SO	Propriété en lecture seule identifiant la taille totale du pool de stockage.
`version`	Valeur numérique	SO	Identifie la version actuelle sur disque du pool. La méthode recommandée de mise à jour des pools consiste à utiliser la commande `zpool upgrade`, bien que cette propriété puisse être utilisée lorsqu'une version spécifique est requise pour des raisons de compatibilité ascendante. Cette propriété peut être définie sur tout numéro compris entre 1 et la version actuelle signalée par la commande `zpool upgrade -v`.

Requête d'état de pool de stockage ZFS

La commande `zpool list` offre plusieurs moyens d'effectuer des demandes sur l'état du pool. Les informations disponibles se répartissent généralement en trois catégories : informations d'utilisation de base, statistiques d'E/S et état de maintenance. Les trois types d'information sur un pool de stockage sont traités dans cette section.

- “Affichage des informations des pools de stockage ZFS” à la page 88
- “Visualisation des statistiques d'E/S des pools de stockage ZFS ” à la page 92
- “Détermination de l'état de maintenance des pools de stockage ZFS” à la page 94

Affichage des informations des pools de stockage ZFS

La commande `zpool list` permet d'afficher les informations de base relatives aux pools.

Affichage des informations concernant tous les pools de stockage ou un pool spécifique

En l'absence d'arguments, la commande `zpool list` affiche les informations suivantes pour tous les pools du système :

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

La sortie de cette commande affiche les informations suivantes :

`NAME` Nom du pool.

`SIZE` Taille totale du pool, égale à la somme de la taille de tous les périphériques virtuels de niveau supérieur.

`ALLOC` Quantité d'espace physique utilisée, c'est-à-dire allouée à tous les jeux de données et métadonnées internes. Notez que cette quantité d'espace disque est différente de celle qui est rapportée au niveau des systèmes de fichiers.

Pour de plus amples informations sur la détermination de l'espace de systèmes de fichiers disponible, reportez-vous à la section "Comptabilisation de l'espace disque ZFS" à la page 32.

`FREE` Quantité d'espace disponible, c'est-à-dire non allouée dans le pool.

`CAP (CAPACITY)` Quantité d'espace disque utilisée, exprimée en tant que pourcentage de l'espace disque total.

`HEALTH` Etat de maintenance actuel du pool.

Pour de plus amples informations sur la maintenance des pools, reportez-vous à la section "Détermination de l'état de maintenance des pools de stockage ZFS" à la page 94.

`ALTROOT` Root de remplacement, le cas échéant.

Pour de plus amples informations sur les pools root de remplacement, reportez-vous à la section "Utilisation de pools root ZFS de remplacement" à la page 290.

Vous pouvez également rassembler des statistiques pour un pool donné en spécifiant le nom du pool. Par exemple :

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

Vous pouvez utiliser l'intervalle `zpool list` et les options de comptage pour rassembler les statistiques d'une période précise. En outre, vous pouvez afficher un horodatage en utilisant l'option `-T`. Par exemple :

```
# 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  -
```

Affichage de statistiques spécifiques à un pool de stockage

L'option `-o` permet d'effectuer une demande concernant des statistiques spécifiques. Cette option permet de générer des rapports personnalisés ou de générer rapidement une liste d'informations pertinentes. Par exemple, pour ne répertorier que le nom et la taille de chaque pool, respectez la syntaxe suivante :

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

Les noms de colonne correspondent aux propriétés répertoriées à la section "Affichage des informations concernant tous les pools de stockage ou un pool spécifique" à la page 89.

Script de sortie du pool de stockage ZFS

La sortie par défaut de la commande `zpool list` a été conçue pour améliorer la lisibilité. Elle n'est pas facile à utiliser en tant que partie d'un script shell. Pour faciliter l'utilisation de la commande dans le cadre de la programmation, l'option `-H` permet de supprimer les en-têtes de colonnes et de séparer les champs par des onglets plutôt que par des espaces. La syntaxe suivante permet d'obtenir la liste des noms de pool du système :

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

Voici un autre exemple :

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

Affichage de l'historique des commandes du pool de stockage ZFS

ZFS consigne automatiquement les commandes `zfs` et `zpool` ayant pour effet de modifier les informations d'état du pool. Cette information peut être affichée à l'aide de la commande `zpool history`.

Par exemple, la syntaxe suivante affiche la sortie de la commande pour le pool root :

```
# 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
```

Vous pouvez utiliser une sortie similaire sur votre système pour identifier l'ensemble *réel* de commandes ZFS exécutées pour résoudre les conditions d'erreur.

Les caractéristiques de l'historique sont les suivantes :

- Le journal ne peut pas être désactivé.
- Le journal est enregistré en permanence sur disque, c'est-à-dire qu'il est conservé d'une réinitialisation système à une autre.
- Le journal est implémenté en tant que tampon d'anneau. La taille minimale est de 128 Ko. La taille maximale est de 32 Mo.
- Pour des pools de taille inférieure, la taille maximale est plafonnée à 1 % de la taille du pool, la valeur *size* étant déterminée lors de la création du pool.
- Le journal ne nécessite aucune administration, ce qui signifie qu'il n'est pas nécessaire d'ajuster la taille du journal ou de modifier son emplacement.

Pour identifier l'historique des commandes d'un pool de stockage spécifique, utilisez une syntaxe similaire à la suivante :

```
# 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
```

Utilisez l'option `-l` pour afficher un format détaillé comprenant le nom d'utilisateur, le nom de l'hôte et la zone dans laquelle l'opération a été effectuée. Par exemple :

```
# 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]
```

L'option `-i` permet d'afficher des informations relatives aux événements internes utilisables pour établir des diagnostics. Par exemple :

```
# 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 ...
```

Visualisation des statistiques d'E/S des pools de stockage ZFS

La commande `zpool iostat` permet d'effectuer une demande de statistiques d'E/S pour un pool ou des périphériques virtuels spécifiques. Cette commande est similaire à la commande `iostat`. Elle permet d'afficher un instantané statique de toutes les activités d'E/S, ainsi que les statistiques mises à jour pour chaque intervalle spécifié. Les statistiques suivantes sont rapportées :

`alloc capacity`
: Capacité utilisée, c'est-à-dire quantité de données actuellement stockées dans le pool ou le périphérique. Cette quantité diffère quelque peu de la quantité d'espace disque disponible pour les systèmes de fichiers effectifs en raison de détails d'implémentation interne.

  Pour de plus amples informations sur la différence entre l'espace de pool et l'espace de jeux de données, reportez-vous à la section "Comptabilisation de l'espace disque ZFS" à la page 32.

`free capacity`
: Capacité disponible, c'est-à-dire quantité d'espace disque disponible dans le pool ou le périphérique. Comme la statistique `used`, cette quantité diffère légèrement de la quantité d'espace disque disponible pour les jeux de données.

`read operations`	Nombre d'opérations de lecture d'E/S envoyées au pool ou au périphérique, y compris les demandes de métadonnées.
`write operations`	Nombre d'opérations d'écriture d'E/S envoyées au pool ou au périphérique.
`read bandwidth`	Bande passante de toutes les opérations de lecture (métadonnées incluses), exprimée en unités par seconde.
`write bandwidth`	Bande passante de toutes les opérations d'écriture, exprimée en unités par seconde.

Liste des statistiques d'E/S à l'échelle du pool

Sans options, la commande `zpool iostat` affiche les statistiques accumulées depuis l'initialisation pour tous les pools du système. Par exemple :

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

Comme ces statistiques sont cumulatives depuis l'initialisation, la bande passante peut sembler basse si l'activité du pool est relativement faible. Vous pouvez effectuer une demande pour une vue plus précise de l'utilisation actuelle de la bande passante en spécifiant un intervalle. Par exemple :

```
# 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
```

Dans l'exemple ci-dessus, la commande affiche les statistiques d'utilisation pour le pool `tank` toutes les deux secondes, jusqu'à ce que vous saisissiez Ctrl-C. Vous pouvez également spécifier un argument `count` supplémentaire pour entraîner l'interruption de la commande une fois le nombre spécifié d'itérations atteint.

Par exemple, `zpool iostat 2 3` imprimerait un résumé toutes les deux secondes pour trois itérations, pendant six secondes. S'il n'y a qu'un pool unique, les statistiques s'affichent sur des lignes consécutives. S'il existe plusieurs pools, une ligne pointillée supplémentaire délimite chaque itération pour fournir une séparation visuelle.

Liste des statistiques d'E/S des périphériques virtuels

Outre les statistiques d'E/S à l'échelle du pool, la commande `zpool iostat` permet d'afficher des statistiques d'E/S pour des périphériques virtuels. Ainsi, vous pouvez identifier les périphériques anormalement lents ou consulter la répartition d'E/S générées par ZFS. Pour effectuer une demande relative à la disposition complète des périphériques virtuels, ainsi que l'ensemble des statistiques d'E/S, utilisez la commande `zpool iostat -v`. Par exemple :

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

Lors de la visualisation des statistiques d'E/S des périphériques virtuels, vous devez prendre en compte deux points importants :

- Dans un premier temps, les statistiques d'utilisation de l'espace disque sont uniquement disponibles pour les périphériques virtuels de niveau supérieur. L'allocation d'espace disque entre les périphériques virtuels RAID-Z et en miroir est spécifique à l'implémentation et ne s'exprime pas facilement en tant que chiffre unique.
- De plus, il est possible que les chiffres s'additionnent de façon inattendue. En particulier, les opérations au sein des périphériques RAID-Z et mis en miroir ne sont pas parfaitement identiques. Cette différence se remarque particulièrement après la création d'un pool, car une quantité significative d'E/S est réalisée directement sur les disques dans le cadre de la création du pool, qui n'est pas comptabilisée au niveau du miroir. Ces chiffres s'égalisent graduellement dans le temps. Cependant, les périphériques hors ligne, ne répondant pas, ou en panne peuvent également affecter cette symétrie.

Vous pouvez utiliser les mêmes options (interval et count) lorsque vous étudiez les statistiques de périphériques virtuels.

Détermination de l'état de maintenance des pools de stockage ZFS

ZFS offre une méthode intégrée pour examiner la maintenance des pools et des périphériques. La maintenance d'un pool se détermine par l'état de l'ensemble de ses périphériques. La

commande `zpool status` permet d'afficher ces informations d'état. En outre, les défaillances potentielles des pools et des périphériques sont rapportées par la commande `fmd`, s'affichent dans la console système et sont consignées dans le fichier `/var/adm/messages`.

Cette section décrit les méthodes permettant de déterminer la maintenance des pools et des périphériques. Ce chapitre n'aborde cependant pas les méthodes de réparation ou de récupération de pools en mauvais état de maintenance. Pour plus d'informations sur le dépannage et la récupération des données, reportez-vous au Chapitre 10, “Dépannage d'Oracle Solaris ZFS et récupération de pool”.

L'état de maintenance d'un pool est décrit par un des quatre états :

`DEGRADED`
: Pool avec un ou plusieurs périphériques défectueux, mais les données sont toujours disponibles grâce à la configuration redondante.

`ONLINE`
: Pool dont tous les périphériques fonctionnent normalement.

`SUSPENDED`
: Pool attendant la restauration de la connectivité de périphérique. Un pool `SUSPENDED` reste en état wait jusqu'à ce que le problème du périphérique soit résolu.

`UNAVAIL`
: Pool avec des métadonnées altérées, ou des périphériques non disponibles, et pas assez de répliques pour continuer de fonctionner.

Chaque périphérique de pool peut se trouver dans l'un des états suivants :

`DEGRADED` : Le périphérique virtuel a connu un panne. Toutefois, il continue de fonctionner. Cet état est le plus commun lorsqu'un miroir ou un périphérique RAID-Z a perdu un ou plusieurs périphériques le constituant. La tolérance de pannes du pool peut être compromise dans la mesure où une défaillance ultérieure d'un autre périphérique peut être impossible à résoudre.

`OFFLINE` : Le périphérique a été mis hors ligne explicitement par l'administrateur.

`ONLINE` : Le périphérique ou le périphérique virtuel fonctionne normalement. Même si certaines erreurs transitoires peuvent encore survenir, le périphérique fonctionne correctement.

`REMOVED` : Le périphérique a été retiré alors que le système était en cours d'exécution. La détection du retrait d'un périphérique dépend du matériel et n'est pas pris en charge sur toutes les plates-formes.

`UNAVAIL` : L'ouverture du périphérique ou du périphérique virtuel est impossible. Dans certains cas, les pools avec des périphériques en état `UNAVAIL` s'affichent en mode `DEGRADED`. Si un périphérique de niveau supérieur est en état `UNAVAIL`, aucun élément du pool n'est accessible.

La maintenance d'un pool est déterminée à partir de celle de l'ensemble de ses périphériques virtuels. Si l'état de tous les périphériques virtuels est `ONLINE`, l'état du pool est également `ONLINE`. Si l'état d'un des périphériques virtuels est `DEGRADED` ou `UNAVAIL`, l'état du pool est également `DEGRADED`. Si l'état d'un des périphériques virtuels est `UNAVAIL` ou `OFFLINE`, l'état du pool est également `UNAVAIL` ou `SUSPENDED`. Un pool en état `UNAVAIL` ou `SUSPENDED` est complètement inaccessible. Aucune donnée ne peut être récupérée tant que les périphériques nécessaires n'ont pas été connectés ou réparés. Un pool renvoyant l'état `DEGRADED` continue à être exécuté. Cependant, il se peut que vous ne puissiez pas atteindre le même niveau de redondance ou de capacité de données que s'il se trouvait en ligne.

La commande `zpool status` fournit également des informations détaillées sur les opérations de réargenture et de nettoyage.

- Rapport de progression de la réargenture. Par exemple :

```
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
```

- Rapport de progression du nettoyage. Par exemple :

```
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
```

- Message de fin de la réargenture. Par exemple :

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

- Message de fin du nettoyage. Par exemple :

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

- Message d'annulation du nettoyage en cours. Par exemple :

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

- Les messages de fin de la réargenture et du nettoyage subsistent après plusieurs réinitialisation du système.

Etat de maintenance de base de pool de stockage

Vous pouvez vérifier rapidement l'état de maintenance d'un pool en utilisant la commande `zpool status` comme suit :

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

Il est possible d'examiner des pools spécifiques en spécifiant un nom de pool dans la syntaxe de commande. Tout pool n'étant pas en état `ONLINE` doit être passé en revue pour vérifier tout problème potentiel, comme décrit dans la section suivante.

Etat de maintenance détaillé

Vous pouvez demander un résumé de l'état plus détaillé en utilisant l'option `-v`. Par exemple :

```
# 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
```

Cette sortie affiche une description complète des raisons de l'état actuel du pool, y compris une description lisible du problème et un lien vers un article de connaissances contenant de plus amples informations. Les articles de connaissances donnent les informations les plus récentes vous permettant de résoudre le problème. Les informations détaillées de configuration doivent vous permettre de déterminer les périphériques endommagés et la manière de réparer le pool.

Dans l'exemple précédent, le périphérique `UNAVAIL` devrait être remplacé. Une fois le périphérique remplacé, exécutez la commande `zpool online` pour le remettre en ligne, si nécessaire. Par exemple :

```
# 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
```

La sortie ci-dessus identifie que le périphérique reste dans un état dégradé tant qu'aucune réargenture n'a été effectuée.

Si la propriété `autoreplace` est activée, vous n'êtes pas obligé de mettre en ligne le périphérique remplacé.

Si un périphérique d'un pool est hors ligne, la sortie de commande identifie le pool qui pose problème. Par exemple :

```
# 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
```

Les colonnes `READ` et `WRITE` indiquent le nombre d'erreurs d'E/S détectées dans le périphérique, tandis que la colonne `CKSUM` indique le nombre d'erreurs de somme de contrôle impossible à corriger qui se sont produites sur le périphérique. Ces deux comptes d'erreurs indiquent une défaillance potentielle du périphérique et que des actions correctives sont requises. Si le nombre d'erreurs est non nul pour un périphérique virtuel de niveau supérieur, il est possible que des parties de vos données soient inaccessibles.

Le champ `errors:` identifie toute erreur de données connue.

Dans l'exemple de sortie précédent, le périphérique mis en ligne ne cause aucune erreur de données.

Pour plus d'informations sur le diagnostic et la réparation de pools et de données `UNAVAIL`, reportez-vous au Chapitre 10, “Dépannage d'Oracle Solaris ZFS et récupération de pool”.

Collecte des informations sur l'état du pool de stockage ZFS

Vous pouvez utiliser l'intervalle `zpool status` et les options de comptage pour rassembler des statistiques sur une période précise. En outre, vous pouvez afficher un horodatage en utilisant l'option `-T`. Par exemple :

```
# 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
```

Migration de pools de stockage ZFS

Parfois, il est possible que vous deviez déplacer un pool de stockage d'un système à l'autre. Pour ce faire, les périphériques de stockage doivent être déconnectés du système d'origine et reconnectés au système de destination. Pour accomplir cette tâche, vous pouvez raccorder physiquement les périphériques ou utiliser des périphériques multiport, par exemple les périphériques d'un SAN. Le système de fichiers ZFS permet d'exporter le pool à partir d'un système et de l'importer sur le système de destination, même si l'endianisme de l'architecture des systèmes est différente. Pour plus d'informations sur la réplication ou la migration de systèmes de fichiers d'un pool de stockage à un autre résidant éventuellement sur des systèmes différents, reportez-vous à la section "Envoi et réception de données ZFS" à la page 229.

- "Préparatifs de migration de pool de stockage ZFS" à la page 100
- "Exportation d'un pool de stockage ZFS" à la page 101
- "Définition des pools de stockage disponibles pour importation" à la page 101
- "Importation de pools de stockage ZFS à partir d'autres répertoires" à la page 103
- "Importation de pools de stockage ZFS" à la page 103
- "Récupération de pools de stockage ZFS détruits" à la page 107

Préparatifs de migration de pool de stockage ZFS

Il est conseillé d'exporter les pools de stockage explicitement afin d'indiquer qu'ils sont prêts à la migration. Cette opération vide toute donnée non écrite sur le disque, écrit les données sur le disque en indiquant que l'exportation a été effectuée et supprime toute information sur le pool du système.

Si vous retirez les disques manuellement, au lieu d'exporter le pool explicitement, vous pouvez toujours importer le pool résultant dans un autre système. Cependant, vous pourriez perdre les dernières secondes de transactions de données et le pool s'affichera alors comme `UNAVAIL` sur le système d'origine dans la mesure où les périphériques ne sont plus présents. Par défaut, le système de destination refuse d'importer un pool qui n'a pas été exporté implicitement. Cette condition est nécessaire car elle évite les importations accidentelles d'un pool actif composé de stockage connecté au réseau toujours en cours d'utilisation sur un autre système.

Exportation d'un pool de stockage ZFS

La commande `zpool export` permet d'exporter un pool. Par exemple :

```
# zpool export tank
```

La commande tente de démonter tout système de fichiers démonté au sein du pool avant de continuer. Si le démontage d'un des systèmes de fichiers est impossible, vous pouvez le forcer à l'aide de l'option `-f`. Par exemple :

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

Une fois la commande exécutée, le pool `tank` n'est plus visible sur le système.

Si les périphériques ne sont pas disponibles lors de l'export, les périphériques ne peuvent pas être identifiés comme étant exportés sans défaut. Si un de ces périphériques est connecté ultérieurement à un système sans aucun des périphériques en mode de fonctionnement, il s'affiche comme étant "potentiellement actif".

Si des volumes ZFS sont utilisés dans le pool, ce dernier ne peut pas être exporté, même avec l'option `-f`. Pour exporter un pool contenant un volume ZFS, vérifiez au préalable que tous les utilisateurs du volume ne sont plus actifs.

Pour de plus amples informations sur les volumes ZFS, reportez-vous à la section "Volumes ZFS" à la page 281.

Définition des pools de stockage disponibles pour importation

Une fois le pool supprimé du système (soit par le biais d'une exportation explicite, soit par le biais d'une suppression forcée des périphériques), vous pouvez connecter les périphériques au système cible. Le système de fichiers ZFS peut gérer des situations dans lesquelles seuls certains périphériques sont disponibles. Cependant, pour migrer correctement un pool, les périphériques doivent fonctionner correctement. En outre, il n'est pas nécessaire que les périphériques soient connectés sous le même nom de périphérique. ZFS détecte tout périphérique déplacé ou renommé et ajuste la configuration de façon adéquate. Pour connaître les pools disponibles, exécutez la commande `zpool import` sans option. Par exemple :

```
# 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
```

Dans cet exemple, le pool `tank` est disponible pour être importé dans le système cible. Chaque pool est identifié par un nom et un identifiant numérique unique. Si plusieurs pools à importer portent le même nom, vous pouvez utiliser leur identifiant numérique afin de les distinguer.

Tout comme la sortie de la commande `zpool status`, la sortie de la commande `zpool import` se rapporte à un article de connaissances contenant les informations les plus récentes sur les procédures de réparation pour les problèmes qui empêchent l'importation d'un pool. Dans ce cas, l'utilisateur peut forcer l'importation du pool. Cependant, l'importation d'un pool en cours d'utilisation par un autre système au sein d'un réseau de stockage peut entraîner une altération des données et des erreurs graves si les deux systèmes tentent d'écrire dans le même stockage. Si certains périphériques dans le pool ne sont pas disponibles, mais que des données redondantes suffisantes sont disponibles pour obtenir un pool utilisable, le pool s'affiche dans l'état `DEGRADED`. Par exemple :

```
# 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
```

Dans cet exemple, le premier disque est endommagé ou manquant, mais il est toujours possible d'importer le pool car les données mises en miroir restent accessibles. Si le nombre de périphériques non disponibles est trop important, l'importation du pool est impossible.

Dans cet exemple, deux disques manquent dans un périphérique virtuel RAID-Z, ce qui signifie que les données redondantes disponibles ne sont pas suffisantes pour reconstruire le pool. Dans certains cas, les périphériques présents ne sont pas suffisants pour déterminer la configuration complète. Dans ce cas, ZFS ne peut pas déterminer quels autres périphériques faisaient partie du pool, mais fournit autant d'informations que possible sur la situation. Par exemple :

```
# 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.
```

Importation de pools de stockage ZFS à partir d'autres répertoires

Par défaut, la commande `zpool import` ne recherche les périphériques que dans le répertoire `/dev/dsk`. Si les périphériques existent dans un autre répertoire, ou si vous utilisez des pools sauvegardés dans des fichiers, utilisez l'option `-d` pour effectuer des recherches dans d'autres répertoires. Par exemple :

```
# zpool create dozer mirror /file/a /file/b
# zpool export dozer
# zpool import -d /file
  pool: dozer
    id: 7318163511366751416
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        dozer        ONLINE
          mirror-0   ONLINE
            /file/a  ONLINE
            /file/b  ONLINE
# zpool import -d /file dozer
```

Si les périphériques se trouvent dans plusieurs répertoires, vous pouvez utiliser plusieurs options `-d`.

Importation de pools de stockage ZFS

Une fois le pool identifié pour l'importation, vous pouvez l'importer en spécifiant son nom ou son identifiant numérique en tant qu'argument pour la commande `zpool import`. Par exemple :

```
# zpool import tank
```

Si plusieurs pools disponibles possèdent le même nom, vous devez spécifier le pool à importer à l'aide de l'identifiant numérique. Par exemple :

```
# zpool import
  pool: dozer
    id: 2704475622193776801
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        dozer       ONLINE
          c1t9d0    ONLINE

  pool: dozer
    id: 6223921996155991199
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

        dozer       ONLINE
          c1t8d0    ONLINE
# zpool import dozer
cannot import 'dozer': more than one matching pool
import by numeric ID instead
# zpool import 6223921996155991199
```

Si le nom du pool est en conflit avec un nom de pool existant, vous pouvez importer le pool sous un nom différent. Par exemple :

```
# zpool import dozer zeepool
```

Cette commande importe le pool `dozer` exporté sous le nouveau nom `zeepool`. Le nouveau nom de pool est persistant.

Si l'exportation du pool ne s'effectue pas correctement, l'indicateur `-f` est requis par ZFS pour empêcher les utilisateurs d'importer par erreur un pool en cours d'utilisation dans un autre système. Par exemple :

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

Remarque – N'essayez pas d'importer un pool actif sur un seul système vers un autre système. ZFS n'est pas un systèmes de fichiers de cluster natifs, distribués ou parallèles et ne peut pas fournir d'accès simultané à plusieurs hôtes différents.

Les pools peuvent également être importés sous un root de remplacement à l'aide de l'option `-R`. Pour plus d'informations sur les pools root de remplacement, reportez-vous à la section "Utilisation de pools root ZFS de remplacement" à la page 290.

Importation d'un pool avec un périphérique de journalisation manquant

Par défaut, un pool avec un périphérique de journalisation manquant ne peut pas être importé. Vous pouvez utiliser la commande `zpool import -m` pour forcer l'importation d'un pool avec un périphérique de journalisation manquant. Par exemple :

```
# 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
```

Importez le pool avec le périphérique de journalisation manquant. Par exemple :

```
# 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
```

Après avoir connecté le périphérique de journalisation manquant, exécutez la commande `zpool clear` pour effacer les erreurs du pool.

Une récupération similaire peut être tentée avec des périphériques de journalisation mis en miroir manquant. Par exemple :

```
# 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
```

Après avoir connecté les périphériques de journalisation manquant, exécutez la commande `zpool clear` pour effacer les erreurs du pool.

Importation d'un pool en mode lecture seule

Vous pouvez importer un pool en mode lecture seule. Si un pool est tellement endommagé qu'il ne peut pas être accessible, cette fonction peut vous permettre de récupérer les données du pool. Par exemple :

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

Lorsqu'un pool est importé en mode lecture seule, les conditions suivantes s'appliquent :

- Tous les systèmes de fichiers et les volumes sont montés en mode lecture seule.
- Le traitement de la transaction du pool est désactivé. Cela signifie également que les écritures synchrones en attente dans le journal de tentatives ne sont pas lues jusqu'à ce que le pool soit importé en lecture-écriture.
- Les tentatives de définition d'une propriété de pool au cours de l'importation en lecture seule ne sont pas prises en compte.

Un pool en lecture seule peut être redéfini en mode lecture-écriture via l'exportation et l'importation du pool. Par exemple :

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

Importation d'un pool via le chemin d'accès au périphérique

La commande suivante permet d'importer le pool `dpool` en identifiant l'un des périphériques spécifiques du pool, `/dev/dsk/c2t3d0`, dans cet exemple.

```
# 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
```

Même si ce pool est composé de disques entiers, la commande doit inclure l'identificateur de tranche du périphérique concerné.

Récupération de pools de stockage ZFS détruits

La commande `zpool import -D` permet de récupérer un pool de stockage détruit. Par exemple :

```
# 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
```

Dans la sortie de `zpool import`, vous pouvez identifier le pool `tank` comme étant le pool détruit en raison des informations d'état suivantes :

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

Pour récupérer le pool détruit, exécutez la commande `zpool import -D` à nouveau avec le pool à récupérer. Par exemple :

```
# 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
```

Même si l'un des périphériques du pool détruit est indisponible, vous devriez être en mesure de récupérer le pool détruit en incluant l'option `-f`. Dans ce cas, importez le pool défaillant et tentez ensuite de réparer la défaillance du périphérique. Par exemple :

```
# 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
```

Mise à niveau de pools de stockage ZFS

Si certains pools de stockage ZFS proviennent d'une version antérieure de Solaris, vous pouvez mettre les pools à niveau à l'aide de la commande `zpool upgrade` pour bénéficier des fonctions des pools de la version actuelle. De plus, la commande `zpool status` vous avertit lorsque la version de vos pools est plus ancienne. Par exemple :

```
# 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
```

Vous pouvez utiliser la syntaxe suivante afin d'identifier des informations supplémentaires sur une version donnée et sur les versions prises en charge.

```
# 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.
```

Vous pouvez ensuite mettre tous vos pools à niveau en exécutant la commande `zpool upgrade`. Par exemple :

```
# zpool upgrade -a
```

Remarque – Si vous mettez à niveau votre pool vers une version ZFS ultérieure, le pool ne sera pas accessible sur un système qui exécute une version ZFS plus ancienne.

Si vous souhaitez utiliser la console d'administration ZFS dans un système comprenant un pool provenant d'une version précédente de Solaris, veillez préalablement à mettre les pools à niveau. La commande `zpool status` permet de déterminer si une mise à niveau des pools est requise.

◆ ◆ ◆ CHAPITRE 4

Installation et initialisation d'un système de fichiers root ZFS Oracle Solaris

Ce chapitre décrit la procédure d'installation et d'initialisation d'un système de fichiers root ZFS Oracle Solaris. La migration d'un système de fichiers root UFS vers un système de fichiers ZFS à l'aide de la fonction Oracle Solaris Live Upgrade y est également abordée.

Ce chapitre contient les sections suivantes :

- "Installation et initialisation d'un système de fichiers root Oracle Solaris ZFS (présentation) " à la page 112
- "Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS" à la page 113
- "Installation d'un système de fichiers root ZFS (installation initiale d'Oracle Solaris)" à la page 116
- "Création d'un pool root ZFS mis en miroir (post-installation)" à la page 122
- "Installation d'un système de fichiers root ZFS (installation d'archive Flash Oracle Solaris)" à la page 124
- "Installation d'un système de fichiers root ZFS (installation JumpStart)" à la page 128
- "Migration d'un système de fichiers root ZFS ou mise à jour d'un système de fichiers root ZFS (Live Upgrade)" à la page 132
- "Gestion de vos périphériques de swap et de vidage ZFS" à la page 158
- "Initialisation à partir d'un système de fichiers root ZFS" à la page 162
- "Restauration du pool root ZFS ou des instantanés du pool root" à la page 170

Vous trouverez la liste des problèmes connus relatifs à cette version dans les *Notes de version Oracle Solaris 10 1/13*.

Installation et initialisation d'un système de fichiers root Oracle Solaris ZFS (présentation)

Vous pouvez installer et initialiser un système de fichiers root ZFS de l'une des manières suivantes :

- **Installation initiale d'Oracle Solaris (méthode d'installation en mode Texte interactif)**
  - Sélectionnez et installez ZFS en tant que système de fichiers root.
  - Installez une archive Flash ZFS.
- **Fonction Oracle Solaris Live Upgrade**
  - Migrez un système de fichiers root UFS vers un système de fichiers root ZFS.
  - Créez un nouvel environnement d'initialisation dans un nouveau pool root ZFS.
  - Créez ou mettez à jour un environnement d'initialisation dans un pool root ZFS existant.
  - Mettez à niveau un autre environnement d'initialisation (BE) à l'aide d'une archive Flash ZFS.
- **Fonction Oracle Solaris JumpStart**.
  - Créez un profil pour installer automatiquement un système avec un système de fichiers root ZFS.
  - Créez un profil pour installer automatiquement un système avec une archive Flash ZFS.

Une fois qu'un système SPARC ou x86 est installé avec un système de fichiers root ZFS ou migré vers un système de fichiers root ZFS, le système s'initialise automatiquement à partir du système de fichiers root ZFS. Pour plus d'informations sur les modifications apportées à l'initialisation, reportez-vous à la section "Initialisation à partir d'un système de fichiers root ZFS" à la page 162.

Fonctions d'installation de ZFS

Les fonctions d'installation de ZFS suivantes sont disponibles dans cette version de Solaris :

- La fonction d'installation interactive en mode Texte vous permet d'installer un système de fichiers root UFS ou ZFS. Le système de fichiers par défaut est toujours UFS pour cette version. Vous pouvez accéder au programme interactif d'installation en mode Texte de l'une des manières suivantes :
  - SPARC : respectez la syntaxe suivante à partir du DVD d'installation d'Oracle Solaris :

    ```
    ok boot cdrom - text
    ```
  - SPARC : respectez la syntaxe suivante lors d'une initialisation à partir du réseau :

    ```
    ok boot net - text
    ```
  - x86 : sélectionnez la méthode d'installation en mode Texte.
- Un profil JumpStart personnalisé permet d'accéder aux fonctionnalités suivantes :

  - Vous pouvez définir un profil pour la création d'un pool de stockage ZFS et désigner un système de fichiers ZFS amorçable.
  - Vous pouvez définir un profil pour installer une archive Flash d'un pool root ZFS.
- Vous pouvez utiliser la fonction Live Upgrade pour migrer un système de fichiers root UFS vers un système de fichiers root ZFS.
- Vous pouvez configurer un pool root ZFS mis en miroir en sélectionnant deux disques au cours de l'installation. Vous pouvez également créer un pool root ZFS mis en miroir en connectant d'autres disques une fois l'installation terminée.
- Les périphériques de swap et de vidage sont automatiquement créés sur les volumes ZFS dans le pool root ZFS.

Les fonctions d'installation suivantes ne sont pas disponibles dans la présente version :

- La fonction d'installation de l'interface graphique permettant d'installer un système de fichiers root ZFS n'est actuellement pas disponible. Vous devez sélectionner la méthode d'installation en mode Texte pour installer un système de fichiers root ZFS.
- Le programme de mise à niveau standard ne peut pas être utilisé pour mettre à niveau votre système de fichiers root UFS avec un système de fichiers root ZFS.

Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS

Assurez-vous de disposer de la configuration requise suivante avant d'installer un système avec un système de fichiers root ZFS ou avant de migrer d'un système de fichiers root UFS vers un système de fichiers root ZFS.

Version Oracle Solaris requise

Vous pouvez installer et initialiser un système de fichiers root ZFS ou migrer vers un système de fichiers root ZFS de différentes manières :

- Installation d'un système de fichiers root ZFS - disponible à partir de la version 10 10/08 de Solaris.
- Migration d'un système de fichiers root UFS vers un système de fichiers root ZFS avec Live Upgrade : vous devez avoir installé la version Solaris 10 10/08 ou une version supérieure, ou vous devez avoir effectué la mise à niveau vers cette version ou une version ultérieure.

Configuration requise générale relative au pool root ZFS

Les sections suivantes décrivent l'espace requis de pool root ZFS et la configuration requise.

Espace disque requis pour les pools root ZFS

La quantité minimale d'espace disponible requis sur le pool d'un système de fichiers root ZFS est supérieure à celle d'un système de fichiers root UFS car les périphériques de swap et de vidage doivent être distincts dans un environnement root ZFS. Par défaut, le périphériques de swap et le périphériques de vidage ne sont qu'un même périphérique sur un système de fichiers root UFS.

Lorsqu'un système est installé ou mis à niveau avec un système de fichiers root ZFS, la taille de la zone de swap et du périphérique de vidage dépend de la quantité de mémoire physique. L'espace minimum de pool disponible d'un système de fichiers root ZFS amorçable dépend de la quantité de mémoire physique, de l'espace disque disponible et du nombre d'environnements d'initialisation à créer.

Consultez les exigences suivantes en matière d'espace disque pour les pools de stockage ZFS :

- 1 536 Mo de mémoire pour installer un système de fichiers root ZFS ;
- 1 536 Mo de mémoire ou plus recommandés pour optimiser les performances ZFS globales ;
- 16 Go d'espace disque minimum recommandés. L'espace disque est utilisé comme suit :
  - **Zone de swap et périphérique de vidage** : les tailles par défaut des volumes de swap et de vidage créés par les programmes d'installation d'Oracle Solaris sont les suivantes :
    - **Installation initiale** : dans le nouvel environnement d'initialisation ZFS, la taille du volume de swap par défaut correspond à la moitié de la taille de la mémoire physique, de 512 Mo à 2 Go en règle générale. Vous pouvez ajuster la taille du volume de swap au cours de l'installation initiale.
    - La taille par défaut du volume de vidage est calculée par le noyau en fonction des informations `dumpadm` et de la taille de la mémoire physique. Vous pouvez régler la taille du volume de vidage au cours de l'installation initiale.
    - **Live Upgrade** : lors de la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS, la taille de swap par défaut de l'environnement d'initialisation ZFS correspond à la taille du périphérique de swap de l'environnement d'initialisation UFS. La taille de swap par défaut est calculée en additionnant la taille de tous les périphériques de swap de l'environnement d'initialisation UFS. Un volume ZFS de cette taille est ensuite créé dans l'environnement d'initialisation ZFS. Si aucun périphérique de swap n'est défini dans l'environnement d'initialisation UFS, la taille de swap par défaut est définie sur 512 Mo.
    - Dans l'environnement d'initialisation ZFS, la taille du volume de vidage par défaut est définie comme étant égale à la moitié de la taille de la mémoire physique. Celle-ci s'étend également de 512 Mo à 2 Go.

    Vous pouvez ajuster la taille de vos volumes de swap et de vidage, dès lors que ces dernières prennent en charge les opérations du système. Pour plus d'informations, reportez-vous à la section "Ajustement de la taille de vos périphériques de swap et de vidage ZFS" à la page 159.

- **Environnement d'initialisation (BE)** : outre l'espace requis pour une nouvelle zone de swap et un nouveau périphérique de vidage ou les tailles ajustées d'une zone de swap et d'un périphérique de vidage, un environnement d'initialisation ZFS migré à partir d'un environnement d'initialisation UFS nécessite environ 6 Go. Chaque environnement d'initialisation ZFS cloné à partir d'un autre environnement d'initialisation ZFS ne nécessite aucun espace disque supplémentaire ; toutefois, lors de l'application de patchs, la taille de l'environnement d'initialisation peut augmenter. Tous les environnements d'initialisation ZFS d'un même pool root utilisent les mêmes périphériques de swap et de vidage.
- **Composants du SE Oracle Solaris** : tous les sous-répertoires du système de fichiers root faisant partie de l'image du SE doivent se trouver dans le même jeu de données que le système de fichiers root, à l'exception de `/var`. En outre, tous les composants du SE doivent se trouver dans le pool root, à l'exception des périphériques de swap et de vidage.

  Pour plus d'informations sur la modification des périphériques de swap et de vidage par défaut avec Live upgrade, reportez-vous à "Personnalisation des volumes de swap et de vidage ZFS" à la page 161.

  Le répertoire ou le jeu de données `/var` doit être un jeu de données unique. Par exemple, si vous voulez également utiliser Live Upgrade pour modifier ou appliquer un patch à un environnement d'initialisation ZFS ou créer une archive Flash de ce pool, vous ne pouvez pas créer un jeu de données `/var` descendant, tel que `/var/tmp`.

Par exemple, un système disposant de 12 Go d'espace disque risque d'être trop petit pour un environnement ZFS amorçable car chaque périphérique de swap et de vidage nécessite 2 Go et l'environnement d'initialisation ZFS migré à partir de l'environnement d'initialisation UFS nécessite environ 6 Go.

Configuration requise pour le pool root ZFS

Vérifiez la configuration requise suivante pour le pool root ZFS :

- Le pool utilisé comme pool root doit contenir une étiquette SMI. Cette condition est généralement respectée si le pool est créé avec des tranches de disque.
- Le pool doit exister sur une tranche de disque ou sur des tranches de disque qui sont mises en miroir. Si vous tentez d'utiliser une configuration de pool non prise en charge lors d'une migration effectuée par Live Upgrade, un message du type suivant s'affiche :

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

  Pour obtenir une description détaillée des configurations de pool root ZFS prises en charge, reportez-vous à la section "Création d'un pool root ZFS" à la page 53.
- x86 : le disque doit comporter une partition Oracle Solaris `fdisk`. Une partition`fdisk` est créée automatiquement lors de l'installation du système x86. Pour plus d'informations sur les partitions `fdisk` Solaris, reportez-vous à la section "Guidelines for Creating an fdisk Partition" du manuel *System Administration Guide: Devices and File Systems*.

- Les disques désignés comme disques d'initialisation dans un pool root ZFS doivent être inférieurs à 2 To sur les systèmes SPARC tout comme sur les systèmes x86.
- La compression peut être activée sur le pool root uniquement après l'installation de ce dernier. Aucun moyen ne permet d'activer la compression sur un pool root au cours de l'installation. L'algorithme de compression `gzip` n'est pas pris en charge sur les pools root.
- Ne renommez pas le pool root après sa création par une installation initiale ou une fois la migration d'Oracle Solaris Live Upgrade vers un système de fichiers root ZFS effectuée. Si vous renommez le pool root, cela peut empêcher l'initialisation du système.

  En outre, si vous souhaitez utiliser Live upgrade, ne modifiez pas le point de montage par défaut des composants du pool root.
- Si vous souhaitez utiliser Live Upgrade et modifier les périphériques de swap et de vidage, reportez-vous à "Personnalisation des volumes de swap et de vidage ZFS" à la page 161.

Installation d'un système de fichiers root ZFS (installation initiale d'Oracle Solaris)

Dans cette version d'Oracle Solaris, vous pouvez effectuer une installation initiale à l'aide des méthodes suivantes :

- Utilisez le programme d'installation interactif en mode Texte pour effectuer l'installation initiale d'un pool de stockage ZFS contenant un système de fichiers root ZFS amorçable. Si vous disposez d'un pool de stockage ZFS que vous souhaitez utiliser pour votre système de fichiers root ZFS, utilisez Live Upgrade pour migrer votre système de fichiers root UFS existant vers un système de fichiers root ZFS dans un pool de stockage ZFS existant. Pour plus d'informations, reportez-vous à la section "Migration d'un système de fichiers root ZFS ou mise à jour d'un système de fichiers root ZFS (Live Upgrade)" à la page 132.
- Utilisez le programme d'installation interactif en mode Texte pour effectuer l'installation initiale d'un pool de stockage ZFS contenant un système de fichiers root ZFS amorçable à partir d'une archive Flash ZFS.

Avant de lancer l'installation initiale pour créer un pool de stockage ZFS, reportez-vous à la section "Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS" à la page 113.

Si vous décidez de configurer des zones après l'installation initiale d'un système de fichiers root ZFS et si vous prévoyez l'application d'un patch au système ou sa mise à niveau, reportez-vous aux sections "Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)" à la page 142 ou "Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)" à la page 147.

Si votre système contient déjà certains pools de stockage ZFS, le message ci-dessous permet de les reconnaître. Cependant, ces pools restent intacts, à moins que vous ne sélectionniez les disques du pool existant pour créer le pool de stockage.

```
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.
```

Attention – Tous les pools existants dont l'un des disques aura été sélectionné pour le nouveau pool seront détruits.

EXEMPLE 4–1 Installation initiale d'un système de fichiers root ZFS amorçable

Le processus d'installation interactif en mode Texte est le même que dans les versions précédentes d'Oracle Solaris, exception faite d'un message vous invitant à créer un système de fichiers root UFS ou ZFS. UFS demeure dans cette version le système de fichiers par défaut. Si vous sélectionnez un système de fichiers root ZFS, un message vous invite à créer un pool de stockage ZFS. Les étapes à suivre pour installer un système de fichiers root ZFS sont les suivantes :

1. Insérez le média d'installation Oracle Solaris ou initialisez le système à partir d'un serveur d'installation. Sélectionnez ensuite la méthode d'installation interactive en mode Texte pour créer un système de fichiers root ZFS amorçable.
   - SPARC : respectez la syntaxe suivante à partir du DVD d'installation d'Oracle Solaris :

     ```
     ok boot cdrom - text
     ```
   - SPARC : respectez la syntaxe suivante lors d'une initialisation à partir du réseau :

     ```
     ok boot net - text
     ```
   - x86 : sélectionnez la méthode d'installation en mode Texte.

   Vous pouvez également créer une archive Flash ZFS à installer à l'aide de l'une des méthodes suivantes :

   - Installation JumpStart. Pour plus d'informations, reportez-vous à l'Exemple 4–2.
   - Installation initiale. Pour plus d'informations, reportez-vous à l'Exemple 4–3.

   Vous pouvez effectuer une mise à niveau standard pour mettre à niveau un système de fichiers ZFS amorçable existant, mais vous ne pouvez pas utiliser cette option pour créer un nouveau système de fichiers ZFS amorçable. A partir de la version Solaris 10 10/08, vous pouvez migrer un système de fichiers root UFS vers un système de fichiers root ZFS, à condition que la version Solaris 10 10/08 ou une version ultérieure soit déjà installée. Pour plus d'informations sur la migration vers un système de fichiers root ZFS, reportez-vous à la section "Migration d'un système de fichiers root ZFS ou mise à jour d'un système de fichiers root ZFS (Live Upgrade)" à la page 132.
2. Pour créer un système de fichiers root ZFS, sélectionnez l'option ZFS. Par exemple :

EXEMPLE 4–1 Installation initiale d'un système de fichiers root ZFS amorçable *(Suite)*

```
Choose Filesystem Type

  Select the filesystem to use for your Solaris installation


           [ ] UFS
           [X] ZFS
```

3. Une fois que le logiciel à installer est sélectionné, un message vous invite à sélectionner les disques pour créer le pool de stockage ZFS. Cet écran est similaire à celui des versions précédentes.

```
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
```

Vous pouvez sélectionner le ou les disques à utiliser pour le pool root ZFS. Si vous sélectionnez deux disques, une configuration de double disque mis en miroir est définie pour le pool root. Un pool mis en miroir comportant 2 ou 3 disques est optimal. Si vous disposez de 8 disques et que vous les sélectionnez tous, ces 8 disques seront utilisés pour le pool root comme un seul grand miroir. Cette configuration n'est pas optimale. Une autre possibilité consiste à créer un pool root mis en miroir une fois l'installation initiale terminée. La configuration de pool RAID-Z n'est pas prise en charge pour le pool root.

Pour plus d'informations sur la configuration des pools de stockage ZFS, reportez-vous à la section "Fonctions de réplication d'un pool de stockage ZFS" à la page 49.

4. Pour sélectionner 2 disques afin de créer un pool root mis en miroir, utilisez les touches fléchées pour sélectionner le deuxième disque.

Dans l'exemple suivant, les deux disques `c1t0d0` et `c1t1d0` sont sélectionnés en tant que disques de pool root. Les deux disques doivent posséder une étiquette SMI et une tranche 0. Si les disques ne sont pas identifiés par une étiquette SMI ou ne contiennent aucune tranche, vous devez quitter le programme d'installation, utiliser l'utilitaire de `formatage` pour réétiqueter et repartitionner les disques, puis relancer le programme d'installation.

EXEMPLE 4–1 Installation initiale d'un système de fichiers root ZFS amorçable *(Suite)*

```
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
) [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
```

Si la colonne Espace disponible renvoie la valeur 0 Mo, le disque dispose probablement d'une étiquette EFI. Si vous voulez utiliser un disque avec une étiquette EFI, vous devrez quitter le programme d'installation, réétiqueter le disque avec une étiquette SMI à l'aide de la commande `format -e`, puis relancer le programme d'installation.

Si vous ne créez pas de pool root mis en miroir lors de l'installation, vous pouvez facilement en créer un après l'installation. Pour plus d'informations, reportez-vous à la section "Création d'un pool root ZFS mis en miroir (post-installation)" à la page 122.

Une fois que vous avez sélectionné un ou plusieurs disques pour le pool de stockage ZFS, un écran semblable au suivant s'affiche :

```
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. Vous pouvez éventuellement, à partir de cet écran, modifier le nom du pool ZFS, le nom du jeu de données, la taille du pool, ainsi que la taille du périphérique de swap et du périphérique de vidage en déplaçant les touches de contrôle du curseur sur les entrées et en

EXEMPLE 4–1 Installation initiale d'un système de fichiers root ZFS amorçable *(Suite)*

remplaçant les valeurs par défaut par de nouvelles valeurs. Vous pouvez aussi accepter les valeurs par défaut. Vous pouvez également modifier la méthode de création et de montage du système de fichiers `/var`.

Dans cet exemple, le nom de jeu de données root est remplacé par `zfsBE`.

```
             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. Dans cet écran d'installation final, vous pouvez éventuellement modifier le profil d'installation. Par exemple :

```
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. Une fois l'installation terminée, vérifiez les informations du pool de stockage et du système de fichiers ZFS. Par exemple :

```
# 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
```

EXEMPLE 4–1 Installation initiale d'un système de fichiers root ZFS amorçable *(Suite)*

```
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  -
```

L'exemple de sortie de la commande `zfs list` identifie les composants du pool root, notamment le répertoire `rpool/ROOT`, qui n'est pas accessible par défaut.

8. Pour créer un autre environnement d'initialisation (BE) ZFS dans le même pool de stockage, utilisez la commande `lucreate`.

   Dans l'exemple suivant, un nouvel environnement d'initialisation nommé `zfs2BE` est créé. L'environnement d'initialisation actuel s'appelle `zfsBE`, comme l'indique la sortie `zfs list`. Toutefois, tant que le nouvel environnement d'initialisation n'est pas créé, l'environnement d'initialisation actuel n'est pas reconnu par la sortie `lustatus`.

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

   Pour créer un environnement d'initialisation ZFS dans le même pool, utilisez une syntaxe du type suivant :

```
# lucreate -n zfs2BE
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

   La création d'un environnement d'exécution ZFS au sein du même pool utilise les fonctions de clonage et d'instantané ZFS pour créer instantanément l'environnement d'exécution. Pour plus d'informations sur l'utilisation de Live Upgrade pour une migration root ZFS, reportez-vous à la section "Migration d'un système de fichiers root ZFS ou mise à jour d'un système de fichiers root ZFS (Live Upgrade)" à la page 132.

9. Vérifiez ensuite les nouveaux environnements d'initialisation. Par exemple :

EXEMPLE 4–1 Installation initiale d'un système de fichiers root ZFS amorçable *(Suite)*

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      yes    yes       no     -
zfs2BE                     yes      no     no        yes    -
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    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. Pour initialiser un système à partir d'un autre environnement d'initialisation, utilisez la commande `luactivate`.

    - SPARC : utilisez la commande `boot -L` pour identifier les environnements d'initialisation disponibles lorsque le périphérique d'initialisation contient un pool de stockage ZFS.

      Par exemple, sur un système SPARC, utilisez la commande `boot -L` pour afficher une liste d'environnements d'initialisation disponibles. Pour initialiser le système à partir du nouvel environnement d'initialisation `zfs2BE`, sélectionnez l'option 2. Saisissez ensuite la commande `boot -Z` affichée.

```
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 : identifiez l'environnement d'initialisation à initialiser à partir du menu GRUB.

Pour plus d'informations sur l'initialisation d'un système de fichiers ZFS, reportez-vous à la section "Initialisation à partir d'un système de fichiers root ZFS" à la page 162.

▼ Création d'un pool root ZFS mis en miroir (post-installation)

Si vous n'avez pas créé de pool root ZFS mis en miroir lors de l'installation, vous pouvez facilement en créer un après l'installation.

Pour obtenir des informations sur le remplacement d'un disque dans un pool root, reportez-vous à la section "Remplacement d'un disque dans le pool root ZFS" à la page 170.

1 Affichez l'état actuel du pool root.

```
# 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 Connectez un deuxième disque pour configurer un pool root mis en miroir.

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

3 Affichez l'état du pool root pour confirmer la fin de la réargenture.

```
# 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
```

Dans la sortie qui précède, le processus de réargenture n'est pas terminé. La réargenture est terminée lorsque des messages similaires aux suivants s'affichent :

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

4 Assurez-vous que vous pouvez initialiser le système à partir du second disque.

5 Si nécessaire, configurez le système pour qu'il puisse s'initialiser automatiquement à partir du nouveau disque.

- SPARC : utilisez la commande `eeprom` ou `setenv` à partir de la PROM d'initialisation SPARC pour réinitialiser le périphérique d'initialisation par défaut.
- x86 : reconfigurez le BIOS du système.

Installation d'un système de fichiers root ZFS (installation d'archive Flash Oracle Solaris)

A partir de la version Solaris 10 10/09, vous pouvez créer une archive Flash sur un système fonctionnant avec un système de fichiers root UFS ou ZFS. Une archive Flash du pool root ZFS contient l'intégralité de la hiérarchie du pool, à l'exception des volumes de swap et de vidage, ainsi que des jeux de données exclus. Lorsque l'archive Flash est installée, les volumes de swap et de vidage sont créés. Vous pouvez utiliser la méthode d'installation d'archive Flash pour effectuer les opérations suivantes :

- Générer une archive Flash pouvant être utilisée pour installer et initialiser un système avec un système de fichiers root ZFS.
- Effectuer une installation JumpStart ou une installation initiale d'un système clone à l'aide d'une archive Flash ZFS. La création d'une archive Flash ZFS clone l'intégralité du pool root, et non les environnements d'initialisation individuels. Les jeux de données individuels au sein du pool peuvent être exclus à l'aide de l'option `-D` des commandes `flarcreate` et `flar`.

Vérifiez les limitations suivantes avant d'installer un système avec une archive Flash ZFS :

- A partir de la version Oracle Solaris 10 8/11, vous pouvez utiliser l'option d'archive Flash de l'installation interactive pour installer un système avec un système de fichiers root ZFS. En outre, vous pouvez utiliser une archive Flash pour mettre à jour un environnement d'initialisation ZFS à l'aide de la commande `luupgrade`.
  - Un système exécutant la version Solaris 10 9/10 doit ajouter les patchs 124630- 51 (SPARC) ou 124631-51 (x86) pour installer une archive Flash sur un ABE.
  - Le système maître sur lequel vous avez créé l'archive Flash et le système clone sur lequel vous allez installer l'archive Flash doivent être au même niveau de patch du noyau. Par exemple, si vous créez une archive Flash ZFS sur un système exécutant la version Solaris 10 8/11, assurez-vous que le système clone exécute le même niveau de patch de noyau de Solaris 10 8/11. Dans le cas contraire, l'installation de l'archive Flash peut échouer en générant des erreurs de la commande `zfs receive`.
  - Un système maître exécutant la version Solaris 10 9/10 avec des systèmes de fichiers root descendants tels qu'un système de fichiers distinct, `/var` doit être mis à niveau à la version Solaris 10 8/11 avant de créer et d'appliquer l'archive Flash à un ABE. Dans le cas contraire, l'installation de l'archive Flash échouera.
- Vous pouvez uniquement installer une archive Flash sur un système doté de la même architecture que le système sur lequel vous avez créé l'archive Flash ZFS. Par exemple, une archive créée sur un système `sun4v` ne peut pas être installée sur un système `sun4u`.
- Seule une nouvelle installation complète d'une archive Flash ZFS est prise en charge. Vous ne pouvez pas installer une archive Flash différentielle d'un système de fichiers root ZFS, ni installer une archive UFS/ZFS hybride.
- A partir de la version Solaris 10 8/11, vous pouvez utiliser une archive Flash UFS pour installer un système de fichiers root ZFS. Par exemple :

  - Si vous utilisez le mot de passe `pool` dans le profil JumpStart, l'archive Flash UFS s'installe dans un pool root ZFS.

    ```
    pool rpool auto auto auto mirror c0t0d0s0 c0t1d0s0
    ```
  - Lors de l'installation interactive d'une archive Flash UFS, sélectionnez ZFS comme type de système de fichiers.
- Bien que la totalité du pool root, à l'exception des jeux de données exclus explicitement, soit archivée et installée, seul l'environnement d'initialisation ZFS initialisé lors de la création de l'archive peut être utilisé après l'installation de l'archive Flash. Cependant, les pools archivés à l'aide de la commande `-R` *rootdir* de la commande `flarcreate` ou `flar` peuvent être utilisés pour archiver un autre pool root que celui en cours d'initialisation.
- Les options des commandes `flarcreate` et `flar` permettant d'inclure et d'exclure des fichiers individuels ne sont pas prises en charge dans une archive Flash ZFS. Vous pouvez uniquement exclure des jeux de données entiers à partir d'une archive Flash ZFS.
- La commande `flar info` n'est pas prise en charge par une archive Flash ZFS. Par exemple :

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

Une fois un système principal installé avec la version Solaris 10 10/09 ou mis à niveau vers cette version, vous pouvez créer une archive Flash ZFS à utiliser pour installer un système cible. Le processus de base est le suivant :

- Créez l'archive Flash ZFS avec la commande `flarcreate` sur le système principal. Tous les jeux de données du pool root sont inclus dans l'archive Flash ZFS, à l'exception des volumes d'échange et de vidage.
- Créez un profil JumpStart pour inclure les informations d'archive Flash sur le serveur d'installation.
- Installez l'archive Flash ZFS sur le système cible.

Les options d'archive suivantes sont prises en charge lors de l'installation d'un pool root ZFS avec une archive Flash :

- Utilisez la commande `flarcreate` ou `flar` pour créer une archive Flash à partir du pool root ZFS spécifié. Si aucune information n'est spécifiée, une archive Flash du pool root par défaut est créée.
- Utilisez la commande `flarcreate -D` *dataset* pour exclure le jeu de données spécifié de l'archive Flash. Cette option peut être utilisée plusieurs fois pour exclure plusieurs jeux de données.

Une fois que vous avez installé une archive Flash ZFS, le système est configuré comme suit :

- L'ensemble de la hiérarchie du jeu de données qui existait sur le système sur lequel l'archive Flash a été créée est recréé sur le système cible, exception faite des jeux de données spécifiquement exclus au moment de la création de l'archive. Les volumes de swap et de vidage ne sont pas inclus dans l'archive Flash.

- Le pool root possède le même nom que le pool qui a été utilisé pour créer l'archive.
- L'environnement d'initialisation qui était actif au moment de la création de l'archive Flash correspond à l'environnement d'initialisation actif et par défaut sur les systèmes déployés.

EXEMPLE 4–2 Installation d'un système avec une archive Flash ZFS (installation JumpStart)

Une fois le système principal installé ou mis à niveau vers la version Solaris 10 10/09 ou ultérieure, créez une archive Flash du pool root ZFS. Par exemple :

```
# 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.
```

Créez un profil JumpStart comme vous le feriez pour installer un système quelconque sur le système à utiliser en tant que serveur d'installation. Par exemple, le profil suivant est utilisé pour installer l'archive `zfs10upflar`:

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

EXEMPLE 4–3 Installation initiale d'un système de fichiers root ZFS amorçable (installation d'archive Flash)

Vous pouvez installer un système de fichiers root ZFS en sélectionnant l'option d'installation Flash. Cette option suppose qu'une archive Flash ZFS a déjà été créée et qu'elle est disponible.

1. Dans l'écran d'installation interactive de Solaris, sélectionnez l'option F4_Flash.
2. Dans l'écran Reboot After Installation (Réinitialiser après l'installation), sélectionnez l'option de réinitialisation automatique ou l'option de réinitialisation manuelle.
3. Dans l'écran de sélection du type de système de fichiers, sélectionnez ZFS.
4. Dans l'écran Flash Archive Retrieval Method (Méthode de récupération d'archive Flash), sélectionnez la méthode de récupération (HTTP, FTP, NFS, fichier local, bande locale ou périphérique local).

   Par exemple, sélectionnez NFS si l'archive Flash ZFS est partagée à partir d'un serveur NFS.
5. Dans l'écran Flash Archive Addition (Ajout d'une archive Flash), indiquez l'emplacement de l'archive Flash ZFS.

EXEMPLE 4–3 Installation initiale d'un système de fichiers root ZFS amorçable (installation d'archive Flash) *(Suite)*

Par exemple, si l'emplacement est un serveur NFS, identifiez le serveur à partir de son adresse IP, puis spécifiez le chemin d'accès à l'archive Flash ZFS.

```
NFS Location: 12.34.567.890:/export/zfs10upflar
```

6. Dans l'écran de sélection de l'archive Flash, confirmez la méthode de récupération et le nom de l'environnement d'initialisation ZFS.

```
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. Comme lors d'une installation initiale, passez en revue les écrans suivants et sélectionnez les options correspondant à votre configuration :
   - Sélection de disques
   - Préserver les données ?
   - Configurer les paramètres ZFS

     Passez en revue le récapitulatif des informations, puis sélectionnez Continue (Continuer).

     Par exemple :

```
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)
```

     Si l'archive Flash est un flux envoyé par ZFS, les options de système de fichiers `/var` combiné ou distinct ne sont pas proposées. Dans ce cas, c'est la configuration de `/var` sur le système maître qui détermine s'il est combiné ou non.
   - Appuyez sur Continue (Continuer) dans l'écran Mount Remote File Systems? (Monter les systèmes de fichiers à distance ?) .
   - Contrôlez l'écran Profile (Profil) et appuyez sur F4 pour apporter des modifications. Sinon, appuyez sur F2 pour commencer l'installation (Begin Installation).

     Par exemple :

EXEMPLE 4–3 Installation initiale d'un système de fichiers root ZFS amorçable (installation d'archive Flash) *(Suite)*

```
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
```

Installation d'un système de fichiers root ZFS (installation JumpStart)

Vous pouvez créer un profil JumpStart pour installer un système de fichiers root ZFS ou UFS.

Un profil JumpStart spécifique au système de fichiers ZFS doit contenir le nouveau mot-clé `pool`. Le mot-clé `pool` installe un nouveau pool root et un nouvel environnement d'initialisation (BE) est créé par défaut. Vous pouvez fournir le nom de l'environnement d'initialisation et créer un jeu de données `/var` distinct à l'aide des mots-clés `bootenv installbe` et des options `bename` et `dataset`.

Pour plus d'informations sur l'utilisation des fonctions JumpStart, reportez-vous au *Guide d'installation d'Oracle Solaris 10 1/13 : Installations JumpStart*.

Si vous décidez de configurer des zones après l'installation JumpStart d'un système de fichiers root ZFS et si vous prévoyez l'application d'un patch au système ou sa mise à niveau, reportez-vous aux sections "Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)" à la page 142 ou "Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)" à la page 147.

Mots-clés JumpStart pour ZFS

Les mots-clés suivants sont autorisés dans un profil JumpStart spécifique à ZFS :

`auto` Spécifie automatiquement la taille des tranches du pool, du volume d'échange ou de vidage. La taille du disque est contrôlée pour s'assurer que la taille minimale peut être satisfaite. Plus la taille est réduite, plus la taille du pool allouée est grande. Cette taille dépend de certaines contraintes, notamment de la taille des disques, des tranches conservées, etc.

Par exemple, lorsque vous spécifiez `c0t0d0s0`, une tranche de pool root aussi grande que possible est créée si vous spécifiez les mots-clés `all` ou `auto`. Vous pouvez également indiquer une taille de tranche, de volume de swap ou de vidage spécifique.

Le mot-clé `auto` fonctionne de la même façon que le mot-clé `all` lorsqu'il est utilisé avec un pool root ZFS, car les pools ne disposent pas d'espace inutilisé.

`bootenv` Identifie les caractéristiques de l'environnement d'initialisation.

Respectez la syntaxe de mot clé `bootenv` suivante pour créer un environnement root ZFS amorçable :

`bootenv installbe bename` *BE-name* [*dataset mount-point*]

`installbe` Crée et installe un nouvel environnement d'initialisation identifié par l'option `bename` et l'entrée *BE-name*.

`bename` *BE-name* Identifie le *BE-name* à installer.

Si l'option `bename` n'est pas utilisée avec le mot-clé `pool`, un environnement d'initialisation par défaut est créé.

`dataset` *mount-point* Utilisez le mot-clé facultatif `dataset` pour identifier un jeu de données `/var` distinct du jeu de données root. La valeur *mount-point* est actuellement limitée à `/var`. Par exemple, une ligne de syntaxe `bootenv` d'un jeu de données `/var` distinct est du type :

```
bootenv installbe bename zfsroot dataset /var
```

`pool` Définit le pool root à créer. La syntaxe de mot-clé suivante doit être respectée :

`pool` *poolname poolsize swapsize dumpsize vdevlist*

poolname Identifie le nom du pool à créer. Le pool est créé avec la taille de pool (*poolsize*) spécifiée et avec les périphériques physiques spécifiés avec un ou plusieurs périphériques (*vdevlist*). La valeur *poolname* ne doit pas identifier le nom d'un pool existant car celui-ci serait écrasé.

poolsize Spécifie la taille du pool à créer. La valeur peut être `auto` ou `existing`. La valeur `auto` attribue la taille de pool la plus grande

possible en fonction des contraintes, notamment de la taille des disques. L'unité de taille supposée s'exprime en Mo, à moins de spécifier `g` (Go).

swapsize Spécifie la taille du volume de swap à créer. La valeur `auto` signifie que la taille de swap par défaut est utilisée. Vous pouvez spécifier une taille avec une valeur *size*. L'unité de taille supposée s'exprime en Mo, à moins de spécifier l'option `g` (Go).

dumpsize Spécifie la taille du volume de vidage à créer. La valeur `auto` signifie que la taille de vidage par défaut est utilisée. Vous pouvez spécifier une taille avec une valeur *size*. L'unité de taille supposée s'exprime en Mo, à moins de spécifier `g` (Go).

vdevlist Spécifie un ou plusieurs périphériques à utiliser pour créer le pool. Le format de l'option *vdevlist* est identique au format de la commande `zpool create`. A l'heure actuelle, seules les configurations mises en miroir sont prises en charge lorsque plusieurs périphériques sont spécifiés. Les périphériques de l'option *vdevlist* doivent correspondre à des tranches du pool root. La valeur `any` signifie que le logiciel d'installation sélectionne le périphérique approprié.

Vous pouvez mettre en miroir autant de disques que vous le souhaitez, mais la taille du pool créé est déterminée par le plus petit des disques spécifiés. Pour plus d'informations sur la création de pools de stockage mis en miroir, reportez-vous à la section "Configuration de pool de stockage mis en miroir" à la page 49.

Exemples de profils JumpStart pour ZFS

Cette section fournit des exemples de profils JumpStart spécifiques à ZFS.

Le profil suivant effectue une installation initiale spécifiée avec `install_type initial-install` dans un nouveau pool, identifié par `pool newpool`, dont la taille est automatiquement définie sur la taille des disques spécifiés par le mot-clé `auto`. La zone de swap et le périphérique de vidage sont automatiquement dimensionnés par le mot-clé `auto` dans une configuration de disques mis en miroir (mot-clé `mirror` et disques `c0t0d0s0` et `c0t1d0s0`). Les caractéristiques de l'environnement d'initialisation sont définies par le mot-clé `bootenv` afin d'installer un nouvel environnement d'initialisation avec le mot-clé `installbe` ; un environnement d'initialisation nommé `s10-xx` est créé.

```
install_type initial_install
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename s10-xx
```

Le profil suivant effectue une installation initiale avec le mot-clé `install_type` `initial-install` du méta-cluster `SUNWCall` dans un nouveau pool de 80 Go portant le nom `newpool`. Ce pool est créé avec un volume d'échange et un volume de vidage de 2 Go chacun dans une configuration mise en miroir dans l'un des deux périphériques disponibles dont la taille permet de créer un pool de 80 Go. Si deux périphériques de ce type ne sont pas disponibles, l'installation échoue. Les caractéristiques de l'environnement d'initialisation sont définies par le mot-clé `bootenv` afin d'installer un nouvel environnement d'initialisation avec le mot-clé `installbe` ; un environnement (`bename`) portant le nom *s10–xx* est créé.

```
install_type initial_install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename s10-xx
```

La syntaxe d'installation JumpStart permet de conserver ou de créer un système de fichiers UFS sur un disque contenant également un pool root ZFS. Cette configuration n'est pas recommandée pour les systèmes de production. Toutefois, elle peut être utilisée pour les besoins en transition ou migration sur un petit système, tel qu'un ordinateur portable.

Problèmes JumpStart pour ZFS

Tenez compte des problèmes suivants avant de lancer une installation JumpStart d'un système de fichiers root ZFS amorçable.

- Un pool de stockage ZFS existant ne peut pas être utilisé lors d'une installation JumpStart pour créer un système de fichiers root ZFS amorçable. Vous devez créer un pool de stockage ZFS conformément au type de syntaxe suivant :

  ```
  pool rpool 20G 4G 4G c0t0d0s0
  ```

- Vous devez créer le pool avec des tranches de disque plutôt qu'avec des disques entiers, comme décrit à la section "Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS" à la page 113. Par exemple, la syntaxe en gras dans l'exemple suivant n'est pas correcte :

  ```
  install_type initial_install
  cluster SUNWCall
  pool rpool all auto auto mirror c0t0d0 c0t1d0
  bootenv installbe bename newBE
  ```

  La syntaxe en gras dans l'exemple suivant est correcte :

  ```
  install_type initial_install
  cluster SUNWCall
  pool rpool all auto auto mirror c0t0d0s0 c0t1d0s0
  bootenv installbe bename newBE
  ```

Migration d'un système de fichiers root ZFS ou mise à jour d'un système de fichiers root ZFS (Live Upgrade)

Les fonctions d'Oracle Live Upgrade concernant les composants UFS sont toujours disponibles et fonctionnent comme dans les versions précédentes de Solaris.

Voici l'ensemble des fonctions disponibles :

- **Migration d'un environnement d'initialisation UFS vers un environnement d'initialisation ZFS**
  - Lorsque vous migrez un système de fichiers root UFS vers un système de fichiers root ZFS, vous devez désigner un pool de stockage ZFS existant à l'aide de l'option `-p`.
  - Si les composants du système de fichiers root UFS sont répartis sur diverses tranches, ils sont migrés vers le pool root ZFS.
  - Dans la version Oracle Solaris 10 8/11, vous pouvez indiquer un système de fichiers `/var` distinct lors de la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS.
  - Le processus de base de la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS est le suivant :
    1. Si nécessaire, installez les patchs Live Upgrade requis.
    2. Installez une version actuelle d'Oracle Solaris 10 (Solaris 10 10/08 à Oracle Solaris 10 8/11) ou utilisez le programme de mise à niveau standard pour effectuer la mise à niveau depuis une version antérieure d'Oracle Solaris 10 sur tout système SPARC ou x86 pris en charge.
    3. Lorsque vous exécutez la version Solaris 10 10/08 ou une version supérieure, créez un pool de stockage ZFS pour votre système de fichiers root ZFS.
    4. Utilisez Live Upgrade pour migrer votre système de fichiers root UFS vers un système de fichiers root ZFS.
    5. Activez votre environnement d'initialisation ZFS à l'aide de la commande `luactivate`.
- **Patch ou mise à niveau d'un environnement d'initialisation ZFS**
  - Vous pouvez utiliser la commande `luupgrade` pour installer un patch ou mettre à niveau un environnement d'initialisation ZFS existant. Vous pouvez également utiliser `luupgrade` pour mettre à niveau un autre environnement d'initialisation ZFS à l'aide d'une archive Flash ZFS. Pour plus d'informations, reportez-vous à l'Exemple 4–8.
  - Lorsque vous créez un environnement d'initialisation ZFS dans le même pool, Live Upgrade peut faire appel aux fonctions d'instantané et de clonage ZFS. La création d'un environnement d'initialisation est de ce fait plus rapide que dans les versions précédentes.

- **Prise en charge de la migration de zones** : vous migrez un système comportant des zones, mais les configurations prises en charge sont limitées dans la version Solaris 10 10/08. A compter de la version 10 5/09, Solaris prend en charge davantage de configurations de zone. Pour plus d'informations, consultez les sections suivantes :
  - "Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)" à la page 142
  - "Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)" à la page 147

  Pour la migration vers un système sans zones, reportez-vous à la section "Utilisation de Live Upgrade pour migrer ou mettre à jour un système de fichiers root ZFS (sans zones)" à la page 134.

Pour plus d'informations sur l'installation d'Oracle Solaris et sur les fonctions de Live Upgrade, reportez-vous au *Guide d'installation d'Oracle Solaris 10 1/13 : Live Upgrade et planification de la mise à niveau*.

Pour plus d'informations sur la configuration requise de ZFS et de Live Upgrade, reportez-vous à la section "Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS" à la page 113.

Problèmes de migration ZFS avec Live Upgrade

Vérifiez la liste de problèmes suivants avant d'utiliser Live Upgrade pour migrer votre système de fichiers root UFS vers un système de fichiers root ZFS :

- L'option de mise à niveau standard de l'interface graphique du programme d'installation d'Oracle Solaris n'est pas disponible pour la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS. Pour migrer un système de fichiers UFS, vous devez utiliser Live Upgrade.
- Vous devez créer le pool de stockage ZFS à utiliser pour l'initialisation avant de lancer Live Upgrade. En outre, en raison des limitations actuelles de l'initialisation, le pool root ZFS doit être créé avec des tranches plutôt qu'avec des disques entiers. Par exemple :

  ```
  # zpool create rpool mirror c1t0d0s0 c1t1d0s0
  ```

  Avant de créer le nouveau pool, assurez-vous que les disques à utiliser dans le pool portent une étiquette SMI (VTOC) au lieu d'une étiquette EFI. Si le disque a été réétiqueté avec une étiquette SMI, assurez-vous que le processus d'étiquetage n'a pas modifié le schéma de partitionnement. Dans la plupart des cas, toute la capacité du disque doit se trouver dans les tranches destinées au pool root.

- Vous ne pouvez pas utiliser Oracle Solaris Live Upgrade pour créer un environnement d'initialisation UFS à partir d'un environnement d'initialisation ZFS. Si vous migrez votre environnement d'initialisation UFS vers un environnement d'initialisation ZFS tout en conservant votre environnement d'initialisation UFS, vous pouvez initialiser l'un ou l'autre des environnements (UFS ou ZFS).
- Ne renommez pas vos environnements d'initialisation ZFS à l'aide de la commande `zfs rename` car Live Upgrade ne détecte pas les changements de nom. Toute commande utilisée ultérieurement, notamment `ludelete`, échoue. Ne renommez en fait pas vos pools ZFS ni vos systèmes de fichiers ZFS si vous disposez d'environnements d'initialisation que vous souhaitez continuer à utiliser.
- Lorsque vous créez un autre environnement d'initialisation cloné sur l'environnement d'initialisation principal, vous ne pouvez pas utiliser les options `-f`, `-x`, `-y`, `-Y` et `-z` pour inclure ou exclure des fichiers de l'environnement d'initialisation principal. Vous pouvez toutefois vous servir des options d'inclusion et d'exclusion dans les cas suivants :

  ```
  UFS -> UFS
  UFS -> ZFS
  ZFS -> ZFS (different pool)
  ```
- Bien que vous puissiez utiliser Live Upgrade pour mettre à niveau votre système de fichiers root UFS vers un système de fichiers root ZFS, vous ne pouvez pas vous servir de Live Upgrade pour mettre à niveau des systèmes de fichiers partagés ni des systèmes de fichiers qui ne sont pas des systèmes de fichiers root.
- Vous ne pouvez pas utiliser une commande `lu` pour créer ou migrer un système de fichiers root ZFS.
- Si vous souhaitez disposer de votre système de périphérique de swap et de vidage dans un pool non-root, reportez-vous à "Personnalisation des volumes de swap et de vidage ZFS" à la page 161.

Utilisation de Live Upgrade pour migrer ou mettre à jour un système de fichiers root ZFS (sans zones)

Les exemples suivants illustrent la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS et la mise à jour d'un système de fichiers root ZFS.

Si vous migrez un système comportant des zones ou effectuez sa mise à jour, reportez-vous aux sections suivantes :

- "Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)" à la page 142
- "Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)" à la page 147

EXEMPLE 4–4 Utilisation de Live Upgrade pour migrer un système de fichiers root UFS vers un système de fichiers root ZFS

L'exemple suivant montre comment effectuer la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS. L'environnement d'initialisation actuel, `ufsBE`, qui contient un système de fichiers root UFS, est identifié par l'option `-c`. Si vous n'incluez pas l'option `-c`, le nom actuel de l'environnement d'initialisation sera par défaut le nom du périphérique. Le nouvel environnement d'initialisation, `zfsBE`, est identifié par l'option `-n`. L'utilisation de la commande `lucreate` requiert l'existence préalable d'un pool de stockage ZFS.

Pour pouvoir mettre à niveau et initialiser le pool de stockage ZFS, vous devez le créer avec des tranches plutôt que des disques entiers. Avant de créer le nouveau pool, assurez-vous que les disques à utiliser dans le pool portent une étiquette SMI (VTOC) au lieu d'une étiquette EFI. Si le disque a été réétiqueté avec une étiquette SMI, assurez-vous que le processus d'étiquetage n'a pas modifié le schéma de partitionnement. Dans la plupart des cas, toute la capacité du disque doit se trouver dans la tranche destinée au pool root.

```
# zpool create rpool mirror c1t2d0s0 c2t1d0s0
# lucreate -c ufsBE -n zfsBE -p rpool
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <ufsBE>.
Creating initial configuration for primary boot environment <ufsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c1t2d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
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.
```

EXEMPLE 4–4 Utilisation de Live Upgrade pour migrer un système de fichiers root UFS vers un système de fichiers root ZFS *(Suite)*

Une fois l'exécution de la commande `lucreate` terminée, utilisez la commande `lustatus` pour afficher l'état de l'environnement d'initialisation. Par exemple :

```
# 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    -
```

Examinez ensuite la liste des composants ZFS. Par exemple :

```
# 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  -
```

Utilisez ensuite la commande `luactivate` pour activer le nouvel environnement d'initialisation ZFS. Par exemple :

```
# 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.

**********************************************************************
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
```

Réinitialisez ensuite le système afin d'utiliser l'environnement d'initialisation ZFS.

```
# init 6
```

Confirmez que l'environnement d'initialisation ZFS est actif.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
```

EXEMPLE 4–4 Utilisation de Live Upgrade pour migrer un système de fichiers root UFS vers un système de fichiers root ZFS *(Suite)*

```
------------------------- -------- ------ --------- ------ ----------
ufsBE                     yes      no     no        yes    -
zfsBE                     yes      yes    yes       no     -
```

Si vous repassez à l'environnement d'initialisation UFS, vous devez réimporter tous les pools de stockage ZFS créés au cours de l'initialisation de l'environnement d'initialisation ZFS car ils ne sont pas disponibles automatiquement dans l'environnement d'initialisation UFS.

Lorsque l'environnement d'initialisation UFS est obsolète, vous pouvez le supprimer à l'aide de la commande `ludelete`.

EXEMPLE 4–5 Utilisation de Live Upgrade pour créer un environnement d'initialisation ZFS à partir d'un environnement d'initialisation UFS (avec un `/var` distinct)

Dans la version Oracle Solaris 10 8/11, vous pouvez utiliser l'option `lucreate -D` pour indiquer que vous souhaitez qu'un autre système de fichiers `/var` soit créé lors de la migration d'un système de fichiers root UFS vers un système de fichiers root ZFS. Dans l'exemple suivant, l'environnement d'initialisation UFS existant est migré vers un environnement d'initialisation ZFS avec un système de fichiers `/var` distinct.

```
# 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>.
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.
```

EXEMPLE 4–5 Utilisation de Live Upgrade pour créer un environnement d'initialisation ZFS à partir d'un environnement d'initialisation UFS (avec un `/var` distinct) *(Suite)*

```
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
```

Passez en revue les systèmes de fichiers ZFS nouvellement créés. Par exemple :

```
# 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  -
```

EXEMPLE 4–6 Utilisation de Live Upgrade pour créer un environnement d'initialisation ZFS à partir d'un environnement d'initialisation ZFS

La création d'un environnement d'initialisation ZFS à partir d'un environnement d'initialisation ZFS du même pool est très rapide car l'opération fait appel aux fonctions d'instantané et de clonage ZFS. Si l'environnement d'initialisation actuel se trouve dans le même pool ZFS, l'option `-p` est omise.

Si vous disposez de plusieurs environnements d'initialisation ZFS, procédez comme suit pour sélectionner l'environnement d'initialisation à partir duquel vous voulez initialiser le système :

- SPARC : vous pouvez utiliser la commande `boot -L` pour identifier les environnements d'initialisation disponibles. Ensuite, sélectionnez un environnement d'initialisation à partir duquel effectuer l'initialisation en utilisant la commande `boot -Z`.
- x86 : vous pouvez sélectionner un environnement d'initialisation à partir du menu GRUB.

Pour plus d'informations, reportez-vous à l'Exemple 4–12.

```
# lucreate -n zfs2BE
Analyzing system configuration.
```

EXEMPLE 4–6 Utilisation de Live Upgrade pour créer un environnement d'initialisation ZFS à partir d'un environnement d'initialisation ZFS *(Suite)*

```
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.
```

EXEMPLE 4–7 Mise à niveau de votre environnement d'initialisation ZFS (`luupgrade`)

Vous pouvez mettre à niveau votre environnement d'initialisation ZFS à l'aide de packages ou de patchs supplémentaires.

Le processus de base est le suivant :

- Créez un environnement d'initialisation alternatif à l'aide de la commande `lucreate`.
- Activez et initialisez le système à partir de l'environnement d'initialisation alternatif.
- Mettez votre environnement d'initialisation ZFS principal à niveau l'aide de la commande `luupgrade` pour ajouter des packages ou des patchs.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      no     no        yes    -
zfs2BE                     yes      yes    yes       no     -
# luupgrade -p -n zfsBE -s /net/system/export/s10up/Solaris_10/Product SUNWchxge
Validating the contents of the media </net/install/export/s10up/Solaris_10/Product>.
Mounting the BE <zfsBE>.
Adding packages to the BE <zfsBE>.

Processing package instance <SUNWchxge> from </net/install/export/s10up/Solaris_10/Product>

Chelsio N110 10GE NIC Driver(sparc) 11.10.0,REV=2006.02.15.20.41
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.

This appears to be an attempt to install the same architecture and
version of a package which is already installed.  This installation
```

EXEMPLE 4–7 Mise à niveau de votre environnement d'initialisation ZFS (`luupgrade`) *(Suite)*

```
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.
```

Vous pouvez aussi créer un nouvel environnement d'initialisation à mettre à jour avec une version ultérieure d'Oracle Solaris. Par exemple :

```
# luupgrade -u -n newBE -s /net/install/export/s10up/latest
```

L'option `-s` représente l'emplacement d'un mode d'installation Solaris.

EXEMPLE 4–8 Création d'un environnement d'initialisation ZFS avec une archive Flash ZFS (`luupgrade`)

Dans la version Oracle Solaris 10 8/11, vous pouvez utiliser la commande `luupgrade` pour créer un environnement d'initialisation ZFS à partir d'une archive Flash ZFS. Le processus de base est comme décrit ci-après :

1. Créez une archive Flash d'un système maître avec un environnement d'initialisation ZFS.

   Par exemple :

   ```
   master-system# flarcreate -n s10zfsBE /tank/data/s10zfsflar
   Full Flash
   Checking integrity...
   Integrity OK.
   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.
   ```

EXEMPLE 4–8 Création d'un environnement d'initialisation ZFS avec une archive Flash ZFS (`luupgrade`) *(Suite)*

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

2. Mettez l'archive Flash ZFS créée sur le système maître à disposition du système clone.

   Les emplacements d'archives Flash possibles sont un système de fichiers local, HTTP, FTP, NFS, etc.

3. Créez un autre environnement d'initialisation ZFS vide sur le système clone.

   Utilisez l'option `-s -` pour indiquer qu'il s'agit d'un environnement d'initialisation vide destiné à être rempli avec le contenu de l'archive Flash ZFS.

   Par exemple :

```
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. Installez l'archive Flash ZFS dans l'autre environnement d'initialisation.

   Par exemple :

```
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.
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
```

EXEMPLE 4–8 Création d'un environnement d'initialisation ZFS avec une archive Flash ZFS (`luupgrade`) *(Suite)*

```
The Live Flash Install of the boot environment <zfsflashBE> is complete.
```

5. Activez l'environnement d'initialisation ZFS alternatif.

```
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. Réinitialisez le système.

```
clone-system# init 6
```

Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)

Vous pouvez utiliser Live Upgrade pour migrer un système avec zones. Dans Solaris 10 10/08, les configurations prises en charge sont toutefois limitées. Si vous installez Solaris 10 5/09 ou effectuez une mise à niveau vers cette version, un nombre plus important de configurations de zone est pris en charge. Pour plus d'informations, reportez-vous à la section "Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)" à la page 147.

Cette section explique comment configurer et installer un système avec zones de manière à ce qu'il puisse être mis à jour et corrigé avec Live Upgrade. Si vous effectuez la migration vers un système de fichiers root ZFS sans zones, reportez-vous à la section "Utilisation de Live Upgrade pour migrer ou mettre à jour un système de fichiers root ZFS (sans zones)" à la page 134.

Si vous migrez un système avec zones ou configurez un système avec zones dans Solaris 10 10/08, consultez les procédures suivantes :

- "Migration d'un système de fichiers root UFS avec roots de zone sur UFS vers un système de fichiers root ZFS (Solaris 10 10/08)" à la page 143
- "Configuration d'un système de fichiers root ZFS avec roots de zone sur ZFS (Solaris 10 10/08)" à la page 145
- "Mise à niveau ou application de patch à un système de fichiers root ZFS avec roots de zone sur ZFS (Solaris 10 10/08)" à la page 146
- "Résolution de problèmes de point de montage empêchant l'initialisation (Solaris 10 10/08)" à la page 167

Suivez les procédures recommandées pour configurer des zones sur un système avec système de fichiers root ZFS pour vérifier que vous pouvez utiliser Live Upgrade sur ce système.

▼ Migration d'un système de fichiers root UFS avec roots de zone sur UFS vers un système de fichiers root ZFS (Solaris 10 10/08)

La procédure suivante explique comment migrer d'un système de fichiers root UFS comportant des zones installées vers un système de fichiers root ZFS et une configuration de root de zone ZFS pouvant être mis à niveau ou patchés.

Dans les étapes suivantes, le pool porte le nom `rpool` et les noms des environnements d'initialisation débutent par `S10BE*` .

1 Mettez le système à niveau à la version Solaris 10 10/08 si la version Solaris 10 exécutée est antérieure.

Pour plus d'informations sur la mise à niveau d'un système exécutant Solaris 10, reportez-vous au *Guide d'installation d'Oracle Solaris 10 1/13 : Live Upgrade et planification de la mise à niveau*.

2 Créez le pool root.

```
# zpool create rpool mirror c0t1d0 c1t1d0
```

Pour plus d'informations sur la configuration requise pour le pool root, reportez-vous à la section "Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS" à la page 113.

3 Confirmez que les zones de l'environnement UFS sont initialisées.

4 Créez le nouvel environnement d'initialisation ZFS.

```
# lucreate -n s10BE2 -p rpool
```

Cette commande crée des jeux de données dans le pool root pour le nouvel environnement d'initialisation et copie l'environnement d'initialisation actuel (zones incluses) vers ces jeux de données.

5 Activez le nouvel environnement d'initialisation ZFS.

```
# luactivate s10BE2
```

Le système exécute désormais un système de fichiers root ZFS, mais les roots de zone sur UFS se trouvent toujours sur le système de fichiers root UFS. Les étapes suivantes sont nécessaires pour finaliser la migration des zones UFS vers une configuration ZFS prise en charge.

6 Réinitialisez le système.

```
# init 6
```

7 Migrez les zones vers un environnement d'initialisation ZFS.

a. Initialisez les zones.

b. Créez un autre environnement d'initialisation ZFS dans le pool.

```
# lucreate s10BE3
```

c. Activez le nouvel environnement d'initialisation.

```
# luactivate s10BE3
```

d. Réinitialisez le système.

```
# init 6
```

Cette étape vérifie que l'environnement d'initialisation ZFS et les zones ont été initialisés.

8 Résolvez les éventuels problèmes de point de montage.

Etant donné la présence d'un bogue dans Live Upgrade, il se peut que l'environnement d'initialisation inactif ne puisse pas s'initialiser. Ce problème est lié à la présence d'un point de montage non valide dans un jeu de données ZFS ou dans un jeu de données ZFS d'une zone de l'environnement d'initialisation.

a. Contrôlez la sortie `zfs list`.

Vérifiez qu'elle ne contient aucun point de montage temporaire erroné. Par exemple :

```
# 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
```

Le point de montage pour l'environnement d'initialisation root ZFS (`rpool/ROOT/s10up`) doit être `/`.

b. Réinitialisez les points de montage pour l'environnement d'initialisation ZFS et ses jeux de données.

Par exemple :

```
# zfs inherit -r mountpoint rpool/ROOT/s10up
# zfs set mountpoint=/ rpool/ROOT/s10up
```

c. Réinitialisez le système.

Lorsque l'option d'initialisation d'un environnement d'initialisation spécifique est proposée, à l'invite de la PROM OpenBoot ou dans le menu GRUB, sélectionnez l'environnement d'initialisation dont les points de montage viennent d'être corrigés.

▼ Configuration d'un système de fichiers root ZFS avec roots de zone sur ZFS (Solaris 10 10/08)

La procédure suivante explique comment installer un système de fichiers root ZFS et une configuration root de zone ZFS pouvant être mis à niveau ou patchés. Dans cette configuration, les roots de zone ZFS sont créés sous forme de jeux de données ZFS.

Dans les étapes qui suivantes, le pool porte le nom `rpool` et l'environnement d'initialisation actif porte le nom `s10BE`. Le nom du jeu de données de zones peut être tout nom de jeu de données valide. Dans l'exemple suivant, le jeu de données des zones porte le nom `zones`.

1 Installez le système avec un root ZFS en utilisant soit le programme d'installation en mode Texte interactif , soit la méthode d'installation JumpStart.

Selon la méthode d'installation que vous choisissez, reportez-vous à la section "Installation d'un système de fichiers root ZFS (installation initiale d'Oracle Solaris)" à la page 116 ou à la section "Installation d'un système de fichiers root ZFS (installation JumpStart)" à la page 128.

2 Initialisez le système à partir du pool root créé.

3 Créez un jeu de données pour le regroupement des roots de zone.

Par exemple :

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones
```

La définition de la valeur `noauto` pour la propriété `canmount` permet d'éviter que le jeu de données ne soit monté d'une manière autre que par l'action explicite de Live Upgrade et le code de démarrage du système.

4 Montez le jeu de données de zones créé.

```
# zfs mount rpool/ROOT/s10BE/zones
```

Le jeu de données est monté sous `/zones`.

5 Créez et montez un jeu de données pour chaque root de zone.

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones/zonerootA
# zfs mount rpool/ROOT/s10BE/zones/zonerootA
```

6 Définissez les autorisations appropriées sur le répertoire root de la zone.

```
# chmod 700 /zones/zonerootA
```

7 Configurez la zone en indiquant le chemin de zone comme suit :

```
# 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
```

Vous pouvez activer l'initialisation automatique des zones à l'initialisation du système en respectant la syntaxe suivante :

```
zonecfg:zoneA> set autoboot=true
```

8 **Installez la zone.**

```
# zoneadm -z zoneA install
```

9 **Initialisez la zone.**

```
# zoneadm -z zoneA boot
```

▼ Mise à niveau ou application de patch à un système de fichiers root ZFS avec roots de zone sur ZFS (Solaris 10 10/08)

Suivez cette procédure pour mettre à niveau ou patcher un système de fichiers root ZFS comportant des roots de zone. Ces mises à jour peuvent consister soit en une mise à niveau du système, soit en l'application de patchs.

Dans les étapes suivantes, l'environnement d'initialisation mis à niveau ou corrigé porte le nom newBE.

1 **Créez l'environnement d'initialisation à mettre à jour ou à corriger.**

```
# lucreate -n newBE
```

L'environnement d'initialisation existant, y compris toutes les zones, est cloné. Un jeu de données est créé pour chaque jeu de données de l'environnement d'initialisation d'origine. Ils sont créés dans le même pool que le pool root actuel.

2 **Sélectionnez l'une des options suivantes pour mettre à niveau le système ou appliquer les patchs au nouvel environnement d'initialisation :**

- Mettez à niveau le système.

  ```
  # luupgrade -u -n newBE -s /net/install/export/s10up/latest
  ```

  L'option -s représente l'emplacement du média d'installation d'Oracle Solaris.

- Appliquez les patchs au nouvel environnement d'initialisation.

  ```
  # luupgrade -t -n newBE -t -s /patchdir 139147-02 157347-14
  ```

3 **Activez le nouvel environnement d'initialisation.**

```
# luactivate newBE
```

4 **Procédez à l'initialisation à partir de l'environnement d'initialisation que vous venez d'activer.**

```
# init 6
```

5 Résolvez les éventuels problèmes de point de montage.

Etant donné la présence d'un bogue dans Live Upgrade, il se peut que l'environnement d'initialisation inactif ne puisse pas s'initialiser. Ce problème est lié à la présence d'un point de montage non valide dans un jeu de données ZFS ou dans un jeu de données ZFS d'une zone de l'environnement d'initialisation.

a. Contrôlez la sortie `zfs list`.

Vérifiez qu'elle ne contient aucun point de montage temporaire erroné. Par exemple :

```
# 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
```

Le point de montage pour l'environnement d'initialisation root ZFS (`rpool/ROOT/newBE`) doit être `/`.

b. Réinitialisez les points de montage pour l'environnement d'initialisation ZFS et ses jeux de données.

Par exemple :

```
# zfs inherit -r mountpoint rpool/ROOT/newBE
# zfs set mountpoint=/ rpool/ROOT/newBE
```

c. Réinitialisez le système.

Lorsque l'option d'initialisation d'un environnement d'initialisation spécifique est proposée, à l'invite de la PROM OpenBoot ou dans le menu GRUB, sélectionnez l'environnement d'initialisation dont les points de montage viennent d'être corrigés.

Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)

A partir de la version 10 10/08 de Solaris, vous pouvez utiliser la fonctionnalité Oracle Solaris Live Upgrade pour migrer ou mettre à niveau un système avec zones. A partir de la version Solaris 10 5/09, des configurations de zone sparse (entière ou root) supplémentaires sont prises en charge par Live Upgrade.

Cette section décrit comment configurer et installer un système avec zones de manière à ce qu'il soit mis à niveau et corrigé avec Live Upgrade à partir de la version 10 5/09. Si vous effectuez la

migration vers un système de fichiers root ZFS sans zones, reportez-vous à la section “Utilisation de Live Upgrade pour migrer ou mettre à jour un système de fichiers root ZFS (sans zones)” à la page 134.

Tenez compte des points suivants lorsque vous utilisez Oracle Solaris Live Upgrade avec ZFS et les zones à partir de la version Solaris 10 5/09 :

- A partir de la version Solaris 10 5/09, pour utiliser Live Upgrade avec des configurations de zone prises en charge, vous devez au préalable mettre à niveau votre système vers la version Solaris 10 5/09 ou une version supérieure à l'aide du programme de mise à niveau standard.
- Ensuite, avec Live Upgrade, vous pouvez migrer votre système de fichiers root UFS comprenant des roots de zone vers un système de fichiers root ZFS ou mettre à niveau votre système de fichiers root ZFS et vos roots de zone ou leur appliquer un patch.
- Vous ne pouvez pas migrer les configurations de zones non prises en charge à partir d'une version antérieure à Solaris 10 vers la version 10 5/09 de Solaris ou une version supérieure.

A partir de la version 10 5/09 de Solaris, si vous migrez ou configurez un système comportant des zones, vous devez vérifier les informations suivantes :

- “Système de fichiers ZFS pris en charge avec informations de configuration du root de zone (version Solaris 10 5/09 ou supérieure)” à la page 148
- “Création d'un environnement d'initialisation ZFS avec un système de fichiers root ZFS et une zone root (Solaris 10 5/09 ou version ultérieure)” à la page 150
- “Mise à niveau ou correction d'un système de fichiers root ZFS avec roots de zone (Solaris 10 5/09 ou version ultérieure)” à la page 152
- “Migration d'un système de fichiers root UFS avec root de zone vers un système de fichiers root ZFS (Solaris 10 5/09 ou version ultérieure)” à la page 155

Système de fichiers ZFS pris en charge avec informations de configuration du root de zone (version Solaris 10 5/09 ou supérieure)

Vérifiez les configurations de zone prises en charge avant d'utiliser Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones.

- **Migration d'un système de fichiers root UFS vers un système de fichiers root ZFS**. Les configurations de root de zone suivantes sont prises en charge :
  - Dans un répertoire du système de fichiers root UFS
  - Dans le sous-répertoire d'un point de montage, dans le système de fichiers root UFS
  - Dans un système de fichiers root UFS avec un root de zone se trouvant dans le répertoire d'un système de fichiers root UFS ou dans le sous-répertoire d'un point de montage d'un système de fichiers root UFS et un pool ZFS non-root avec un root de zone

  Un système de fichiers root UFS ayant un root de zone comme point de montage n'est pas pris en charge.

- **Migration ou mise à niveau d'un système de fichiers root ZFS**. Les configurations de root de zone suivantes sont prises en charge :
  - Dans un système de fichiers se trouvant dans un pool root ou non-root ZFS. Par exemple, `/zonepool/zones` est acceptable. Dans certains cas, si un système de fichiers n'est pas fourni pour le root de zone avant l'opération Live Upgrade, un système de fichiers pour le root de zone (`zoneds`) est créé par Live Upgrade.
  - Dans un système de fichiers descendant ou le sous-répertoire d'un système de fichiers ZFS, à condition que différents chemins d'accès à la zone ne soient pas imbriqués. Par exemple, `/zonepool/zones/zone1` et `/zonepool/zones/zone1_dir` sont acceptables.

    Dans l'exemple suivant, `zonepool/zones` est un système de fichiers qui contient les roots de zone et `rpool` contient l'environnement d'initialisation ZFS :

    ```
    zonepool
    zonepool/zones
    zonepool/zones/myzone
    rpool
    rpool/ROOT
    rpool/ROOT/myBE
    ```

    Live Upgrade prend des instantanés et clone les zones contenues dans `zonepool` et dans l'environnement d'initialisation `rpool` si vous respectez la syntaxe suivante :

    ```
    # lucreate -n newBE
    ```

    L'environnement d'initialisation `newBE` sous `rpool/root/newBE` est créé. Lorsque ce dernier est activé, l'environnement d'initialisation `newBE` fournit l'accès aux composants `zonepool`.

    Dans l'exemple qui précède, si `/zonepool/zones` était un sous-répertoire et non un autre système de fichiers, Live Upgrade le migrerait comme un composant du pool root, `rpool`.
  - **La configuration de ZFS et de zone suivante n'est pas prise en charge :**
    - Il n'est pas possible d'utiliser Live Upgrade pour créer un autre environnement d'initialisation lorsque l'environnement d'initialisation source comporte une zone non globale avec un chemin de zone défini sur le point de montage d'un système de fichiers de pool de niveau supérieur. Par exemple, si le pool `zonepool` a un système de fichiers monté comme `/zonepool`, il ne peut pas y avoir de zone non globale possédant un chemin de zone défini sur `/zonepool`.
    - N'ajoutez pas d'entrée de système de fichiers pour une zone non globale dans le fichier `/etc/vfstab` de la zone globale. Utilisez plutôt la fonction `zonecfg add fs` pour ajouter un système de fichiers à une zone non globale.
- **Informations de migration ou de mise à niveau de zones pour les environnements UFS et ZFS** : vérifiez les éléments suivants, car ils sont susceptibles d'affecter la migration ou la mise à niveau d'un environnement ZFS ou UFS :

- Si vous avez configuré les zones comme décrit à la section "Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)" à la page 142 dans la version Solaris 10 10/08 et si vous avez effectué une mise à niveau vers la version Solaris 10 5/09 ou une version ultérieure, vous pouvez effectuer une migration vers un système de fichiers root ZFS ou une mise à niveau vers Solaris 10 5/09 à l'aide de Live Upgrade.
- Ne créez aucun root de zone dans les répertoires imbriqués, par exemple : `zones/zone1` et `zones/zone1/zone2`. Sinon, le montage risque d'échouer lors de l'initialisation.

▼ Création d'un environnement d'initialisation ZFS avec un système de fichiers root ZFS et une zone root (Solaris 10 5/09 ou version ultérieure)

Après avoir procédé à l'installation initiale de la version Solaris 10 5/09 ou d'une version supérieure, suivez cette procédure pour créer un système de fichiers root ZFS. Suivez également la procédure suivante si vous avez utilisé la commande `luupgrade` pour mettre à niveau un système de fichiers root ZFS vers la version 10 5/09 ou une version supérieure de Solaris. Un environnement d'initialisation ZFS créé à l'aide de cette procédure peut être mis à niveau ou corrigé à l'aide d'un patch.

Dans la procédure ci-après, l'exemple de système Oracle Solaris 10 9/10 comporte un système de fichiers root ZFS et un jeu de données root de zone dans `/rpool/zones`. Un environnement d'initialisation ZFS nommé `zfs2BE` est créé et peut ensuite être mis à niveau ou corrigé à l'aide d'un patch.

1 Vérifiez les systèmes de fichiers ZFS existants.

```
# 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 Assurez-vous que les zones sont installées et initialisées.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /rpool/zones                   native   shared
```

3 Créez l'environnement d'initialisation ZFS.

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
```

```
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

4 Activez l'environnement d'initialisation ZFS.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      yes    yes       no     -
zfs2BE                     yes      no     no        yes    -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
```

5 Initialisez l'environnement d'initialisation ZFS.

```
# init 6
```

6 Confirmez que les systèmes de fichiers ZFS et les zones sont créés dans le nouvel environnement d'initialisation.

```
# 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
```

▼ Mise à niveau ou correction d'un système de fichiers root ZFS avec roots de zone (Solaris 10 5/09 ou version ultérieure)

Suivez cette procédure pour mettre à niveau ou corriger un système de fichiers root ZFS avec roots de zone sur Solaris 10 5/09 ou version supérieure. Ces mises à jour peuvent consister en une mise à niveau du système ou en l'application de patchs.

Dans les étapes suivantes, `zfs2BE` est le nom de l'environnement d'initialisation mis à niveau ou corrigé.

1 Vérifiez les systèmes de fichiers ZFS existants.

```
# 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 Assurez-vous que les zones sont installées et initialisées.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   5 zfszone          running    /rpool/zones                   native   shared
```

3 Créez l'environnement d'initialisation ZFS à mettre à jour ou à corriger.

```
# 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 Sélectionnez l'une des options suivantes pour mettre à niveau le système ou appliquer les patchs au nouvel environnement d'initialisation :

- Mettez à niveau le système.

  ```
  # luupgrade -u -n zfs2BE -s /net/install/export/s10up/latest
  ```

  L'option `-s` représente l'emplacement du média d'installation d'Oracle Solaris.

  Ce processus peut être très long.

  Pour un exemple complet du processus `luupgrade`, reportez-vous à l'Exemple 4–9.

- Appliquez les patchs au nouvel environnement d'initialisation.

  ```
  # luupgrade -t -n zfs2BE -t -s /patchdir patch-id-02 patch-id-04
  ```

5 Activez le nouvel environnement d'initialisation.

```
# 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 Initialisez le système à partir de l'environnement d'initialisation récemment activé.

```
# init 6
```

Exemple 4–9 Mise à niveau d'un système de fichiers root ZFS avec root de zone vers un système de fichiers root ZFS Oracle Solaris 10 9/10

Dans cet exemple, un environnement d'initialisation ZFS (`zfsBE`), créé sur un système Solaris 10 10/09 avec un système de fichiers root ZFS et un root de zone dans un pool non-root, est mis à niveau vers Solaris 10 9/10. Ce processus peut être long. L'environnement d'initialisation mis à niveau (`zfs2BE`) est ensuite activé. Assurez-vous que les zones sont installées et initialisées avant de tenter la mise à niveau.

Dans cet exemple, le pool `zonepool`, le jeu de données `/zonepool/zones` et la zone `zfszone` sont créés comme suit :

```
# 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
```

▼ Migration d'un système de fichiers root UFS avec root de zone vers un système de fichiers root ZFS (Solaris 10 5/09 ou version ultérieure)

Suivez cette procédure pour migrer un système incluant un système de fichiers root UFS et un root de zone vers la version Solaris 10 5/09 ou version supérieure. Utilisez ensuite Live Upgrade pour créer un environnement d'initialisation ZFS.

Dans la procédure ci-après, l'exemple de nom de l'environnement d'initialisation UFS est `c1t1d0s0`, le root de zone UFS est `zonepool/zfszone` et l'environnement d'initialisation root est `zfsBE`.

1 Mettez le système à niveau vers la version Solaris 10 5/09 ou un version ultérieure si la version Solaris 10 exécutée est antérieure.

Pour plus d'informations sur la mise à niveau d'un système exécutant Solaris 10, reportez-vous au *Guide d'installation d'Oracle Solaris 10 1/13 : Live Upgrade et planification de la mise à niveau*.

2 Créez le pool root.

Pour plus d'informations sur la configuration requise pour le pool root, reportez-vous à la section "Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS" à la page 113.

3 Confirmez que les zones de l'environnement UFS sont initialisées.

```
# zoneadm list -cv
  ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   2 zfszone          running    /zonepool/zones                native   shared
```

4 Créez le nouvel environnement d'initialisation ZFS.

```
# lucreate -c c1t1d0s0 -n zfsBE -p rpool
```

Cette commande crée des jeux de données dans le pool root pour le nouvel environnement d'initialisation et copie l'environnement d'initialisation actuel (zones incluses) vers ces jeux de données.

5 Activez le nouvel environnement d'initialisation ZFS.

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
c1t1d0s0                   yes      no     no        yes    -
zfsBE                      yes      yes    yes       no     -         #
luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.
.
.
.
```

6 Réinitialisez le système.

```
# init 6
```

7 Confirmez que les systèmes de fichiers ZFS et les zones sont créés dans le nouvel environnement d'initialisation.

```
# 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
```

Exemple 4–10 Migration d'un système de fichiers root UFS avec root de zone vers un système de fichiers root ZFS

Dans cet exemple, un système Oracle Solaris 10 9/10 comprenant un système de fichiers root UFS, un root de zone (`/uzone/ufszone`), un pool non-root ZFS (`pool`) et un root de zone (`/pool/zfszone`) est migré vers un système de fichiers root ZFS. Assurez-vous que le pool root ZFS est créé et que les zones sont installées et initialisées avant de tenter la migration.

```
# 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
```

Gestion de vos périphériques de swap et de vidage ZFS

Lors de l'installation initiale d'un SE Solaris ou après la migration Live Upgrade à partir d'un système de fichiers UFS, une zone de swap est créée sur un volume ZFS du pool root ZFS. Par exemple :

```
# swap -l
swapfile                  dev  swaplo  blocks   free
/dev/zvol/dsk/rpool/swap 256,1      16 4194288 4194288
```

Lors de l'installation initiale d'un SE Oracle Solaris ou de l'utilisation de Live Upgrade à partir d'un système de fichiers UFS, un périphérique de vidage est créé sur un volume ZFS du pool root ZFS. En règle générale, un périphérique de vidage ne nécessite aucune administration car il est créé automatiquement lors de l'installation. Par exemple :

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
  Savecore enabled: yes
   Save compressed: on
```

Si vous désactivez et supprimez le périphérique de vidage, vous devez l'activer avec la commande `dumpadm` après sa recréation. Dans la plupart des cas, vous devrez uniquement ajuster la taille du périphérique de vidage à l'aide de la commande `zfs`.

Pour plus d'informations sur les tailles des volumes de swap et de vidage créés par les programmes d'installation, reportez-vous à la section “Configuration requise pour l'installation d'Oracle Solaris et de Live Upgrade pour la prise en charge de ZFS” à la page 113.

La taille des volume de swap et de vidage peut être ajustée pendant et après l'installation. Pour plus d'informations, reportez-vous à la section “Ajustement de la taille de vos périphériques de swap et de vidage ZFS” à la page 159.

Tenez compte des points suivants lorsque vous travaillez avec des périphériques de swap et de vidage ZFS :

- Vous devez utiliser des volumes ZFS distincts pour la zone de swap et le périphérique de vidage.

- L'utilisation d'un fichier swap sur un système de fichiers ZFS n'est actuellement pas prise en charge.
- Pour modifier la zone de swap ou le périphérique de vidage une fois le système installé ou mis à niveau, utilisez les commandes `swap` et `dumpadm` de la même façon que dans les versions précédentes. Pour plus d'informations, reportez-vous au Chapitre 16, "Configuring Additional Swap Space (Tasks)" du manuel *System Administration Guide: Devices and File Systems* et au Chapitre 17, "Gestion des informations sur les pannes système (tâches)" du manuel *Guide d'administration système : Administration avancée*.

Pour de plus amples informations, reportez-vous aux sections suivantes :

- "Ajustement de la taille de vos périphériques de swap et de vidage ZFS" à la page 159
- "Dépannage du périphérique de vidage ZFS" à la page 161

Ajustement de la taille de vos périphériques de swap et de vidage ZFS

Il peut s'avérer nécessaire d'ajuster la taille des périphériques de swap et de vidage après l'installation ou éventuellement de recréer les volumes de swap et de vidage.

- Vous pouvez ajuster la taille de vos volumes de swap et de vidage au cours d'une installation initiale. Pour plus d'informations, reportez-vous à l'Exemple 4–1.
- Vous pouvez créer des volumes de swap et de vidage, ainsi que leur attribuer une taille, avant de procéder à une opération Live Upgrade. Par exemple :

  1. Créez le pool de stockage.

     ```
     # zpool create rpool mirror c0t0d0s0 c0t1d0s0
     ```

  2. Créez le périphérique de vidage.

     ```
     # zfs create -V 2G rpool/dump
     ```

  3. Activez le périphérique de vidage.

     ```
     # 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. Créez votre volume de swap :

     ```
     # zfs create -V 2G rpool/swap
     ```

  5. Vous devez activer la zone de swap lorsqu'un nouveau périphérique de swap est ajouté ou modifié.

     ```
     # swap -a /dev/zvol/dsk/rpool/swap
     ```

6. Ajoutez une entrée pour le volume d'échange dans le fichier `/etc/vfstab`.

Live Upgrade ne redimensionne pas des volumes de swap et de vidage existants.

- Vous pouvez rétablir la propriété `volsize` du périphérique de vidage après l'installation d'un système. Par exemple :

```
# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME        PROPERTY  VALUE       SOURCE
rpool/dump  volsize   2G          -
```

- Si la zone de swap en cours n'est pas actuellement utilisée, vous pouvez redimensionner la taille du volume de swap en cours, mais vous devez réinitialiser le système pour constater l'augmentation de la taille de l'espace de swap.

```
# 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
```

- Vous pouvez essayer de redimensionner le volume de swap mais il est préférable de supprimer le périphérique de swap. Ensuite, recréez-le. Par exemple :

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs create -V 2g rpool/swap
# swap -a /dev/zvol/dsk/rpool/swap
```

- Vous pouvez ajuster la taille des volumes de swap et de vidage d'un profil JumpStart à l'aide d'une syntaxe de profil du type suivant :

```
install_type initial_install
cluster SUNWCXall
pool rpool 16g 2g 2g c0t0d0s0
```

Dans ce profil, les deux entrées 2g définissent respectivement la taille du volume de swap et celle du volume de vidage sur 2 Go.

- Si vous avez besoin de plus d'espace de swap sur un système déjà installé, il suffit d'ajouter un autre volume de swap. Par exemple :

```
# zfs create -V 2G rpool/swap2
```

Activez ensuite le nouveau volume de swap. Par exemple :

```
# 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
```

Ajoutez ensuite une entrée pour le deuxième volume de swap dans le fichier `/etc/vfstab` .

Personnalisation des volumes de swap et de vidage ZFS

Si vous supprimez la valeur par défaut des volumes de swap et de vidage et que vous les recréez-dans un pool non-root (données), gardez à l'esprit les conseils suivants :

- Si vous souhaitez créer des périphériques de swap et de vidage dans un pool non-root, ne créez pas des volumes de swap et de vidage dans un pool RAIDZ. Si un pool inclut les volumes de swap et de vidage, il doit s'agir d'un pool à un seul disque ou d'un pool mis en miroir.
- Si vous utilisez Live Upgrade pour mettre à jour votre système, utilisez l'option `-P` pour conserver le périphérique de vidage du PBE à l'ABE. Par exemple :

  ```
  # lucreate -n newBE -P
  ```

Dépannage du périphérique de vidage ZFS

Vérifiez les éléments suivants si vous rencontrez des problèmes lors de la capture d'un vidage sur incident du système ou lors du redimensionnement du périphérique de vidage.

- Si un vidage sur incident n'a pas été automatiquement créé, vous pouvez utiliser la commande `savecore` pour enregistrer le vidage sur incident.
- Lorsque vous installez un système de fichiers root ZFS ou lorsque vous effectuez une migration vers un système de fichiers root ZFS pour la première fois, un volume de vidage est automatiquement créé. Dans la plupart des cas, vous devez uniquement ajuster la taille par défaut du volume de vidage si celle-ci est trop petite. Par exemple, vous pouvez augmenter la taille du volume de vidage jusqu'à 40 Go sur un système contenant une quantité de mémoire importante comme suit :

  ```
  # zfs set volsize=40G rpool/dump
  ```

  Le redimensionnement d'un volume de vidage de grande ampleur peut prendre un certain temps.

  Si, pour une raison quelconque, vous devez activer un périphérique de vidage après l'avoir créé manuellement, utilisez une syntaxe semblable à la suivante :

  ```
  # dumpadm -d /dev/zvol/dsk/rpool/dump
        Dump content: kernel pages
         Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
  Savecore directory: /var/crash/t2000
    Savecore enabled: yes
  ```

- Un système avec 128 Go de mémoire ou plus peut nécessiter un périphérique de vidage plus important que celui créé par défaut. Si le périphérique de vidage est trop petit pour capturer un vidage sur incident existant, un message semblable au suivant s'affiche :

```
# 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
```

Pour plus d'informations sur la taille des périphériques de swap et de vidage, reportez-vous à la section "Planning for Swap Space" du manuel *System Administration Guide: Devices and File Systems*.

- Vous ne pouvez actuellement pas ajouter de périphérique de vidage à un pool avec plusieurs périphériques de niveau supérieur. Un message similaire à celui figurant ci-dessous s'affiche :

```
# dumpadm -d /dev/zvol/dsk/datapool/dump
dump is not supported on device '/dev/zvol/dsk/datapool/dump': 'datapool' has multiple top level vdevs
```

Ajoutez le périphérique de vidage au pool root. Ce dernier ne peut pas contenir plusieurs périphériques de niveau supérieur.

Initialisation à partir d'un système de fichiers root ZFS

Les systèmes SPARC et les systèmes x86 utilisent le nouveau type d'initialisation à l'aide d'une archive d'initialisation, qui est une image de système de fichiers contenant les fichiers requis pour l'initialisation. Lorsque vous initialisez un système à partir d'un système de fichiers root ZFS, les noms de chemin de l'archive d'initialisation et du fichier noyau sont résolus dans le système de fichiers root sélectionné pour l'initialisation.

Lorsque vous réinitialisez un système en vue d'une installation, un disque RAM est utilisé pour le système de fichiers root tout au long du processus d'installation.

L'initialisation à partir d'un système de fichiers ZFS diffère de celle à partir d'un système de fichiers UFS car avec ZFS, un spécificateur de périphérique d'initialisation identifie un pool de stockage par opposition à un seul système de fichiers root. Un pool de stockage peut contenir plusieurs *jeux de données amorçables* ou systèmes de fichiers root ZFS. Lorsque vous initialisez un système à partir de ZFS, vous devez spécifier un périphérique d'initialisation et un système de fichiers root contenu dans le pool qui a été identifié par le périphérique d'initialisation.

Par défaut, le jeu de données sélectionné pour l'initialisation est identifié par la propriété `bootfs` du pool. Cette sélection par défaut peut être remplacée en spécifiant un jeu de données amorçable alternatif à l'aide de la commande `boot -Z`.

Initialisation à partir d'un disque alternatif d'un pool root ZFS mis en miroir

Vous pouvez créer un pool root ZFS mis en miroir lors de l'installation du système ou pouvez connecter un disque pour créer un pool root ZFS mis en miroir après l'installation. Pour plus d'informations, consultez les références suivantes :

- "Installation d'un système de fichiers root ZFS (installation initiale d'Oracle Solaris)" à la page 116
- "Création d'un pool root ZFS mis en miroir (post-installation)" à la page 122

Consultez les problèmes connus suivants relatifs aux pools root ZFS mis en miroir :

- Si vous remplacez un disque de pool root en utilisant la commande `zpool replace`, vous devez installer les informations d'initialisation sur le nouveau disque à l'aide de la commande `installboot` ou `installgrub`. Si vous créez un pool root ZFS mis en miroir à l'aide de la méthode d'installation initiale ou si vous utilisez la commande `zpool attach` pour connecter un disque au pool root, cette étape n'est pas nécessaire. La syntaxe des commandes `installboot` et `installgrub` suit le modèle suivant :
  - 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
    ```
- Vous pouvez effectuer l'initialisation à partir de divers périphériques d'un pool root ZFS mis en miroir. Selon la configuration matérielle, la mise à jour de la PROM ou du BIOS peut s'avérer nécessaire pour spécifier un périphérique d'initialisation différent.

  Vous pouvez par exemple effectuer l'initialisation à partir de l'un des deux disques (`c1t0d0s0` ou `c1t1d0s0`) du pool suivant :

  ```
  # 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 : indiquez le disque alternatif à l'invite `ok`. Par exemple :

  ```
  ok boot /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
  ```

  Une fois le système réinitialisé, confirmez le périphérique d'initialisation actif. Par exemple :

```
SPARC# prtconf -vp | grep bootpath
        bootpath:  '/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a'
```

- x86 : sélectionnez un autre disque dans le pool root ZFS mis en miroir à partir du menu BIOS correspondant.

  Respectez ensuite une syntaxe semblable à la suivante pour confirmer l'initialisation à partir du disque de rechange :

```
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 : initialisation à partir d'un système de fichiers root ZFS

Sur un système SPARC avec environnements d'initialisation ZFS multiples, vous pouvez initialiser à partir de tout environnement d'initialisation en utilisant la commande `luactivate`.

Au cours de l'installation du SE Oracle Solaris et de la procédure Live Upgrade, le système de fichiers root ZFS par défaut est automatiquement désigné avec la propriété `bootfs`.

Un pool peut contenir plusieurs jeux de données amorçables. Par défaut, l'entrée du jeu de données amorçable figurant dans le fichier */pool-name*`/boot/menu.lst` est identifiée par la propriété `bootfs` du pool. Cependant, une entrée `menu.lst` peut contenir une commande `bootfs` qui spécifie un jeu de données alternatif du pool. Le fichier `menu.lst` peut ainsi contenir les entrées de plusieurs systèmes de fichiers root du pool.

Lorsqu'un système est installé avec un système de fichiers root ZFS ou est migré vers un système de fichiers root ZFS, une entrée du type suivant est ajoutée au fichier `menu.lst` :

```
title zfsBE
bootfs rpool/ROOT/zfsBE
title zfs2BE
bootfs rpool/ROOT/zfs2BE
```

Lorsqu'un nouvel environnement d'initialisation est créé, le fichier `menu.lst` est mis à jour automatiquement.

Sur un système SPARC, deux options d'initialisation ZFS sont disponibles :

- Une fois l'environnement d'initialisation activé, vous pouvez utiliser la commande d'initialisation `-L` pour afficher une liste de jeux de données amorçable contenus dans un pool ZFS. Vous pouvez ensuite sélectionner un des jeux de données amorçable de la liste. Des instructions détaillées permettant d'initialiser ce jeu de données s'affichent. Vous pouvez initialiser le jeu de données sélectionné en suivant ces instructions.

- Vous pouvez utiliser la commande `-Z` *dataset* pour initialiser un jeu de données ZFS spécifique.

EXEMPLE 4–11 SPARC : Initialisation à partir d'un environnement d'initialisation ZFS spécifique

Si vous disposez de plusieurs environnements d'initialisation ZFS dans un pool de stockage ZFS situé sur le périphérique d'initialisation de votre système, vous pouvez utiliser la commande `luactivate` pour spécifier un environnement d'initialisation par défaut.

Par exemple, la sortie `lustatus` suivante indique que deux environnements d'initialisation ZFS sont disponibles :

```
# lustatus
Boot Environment           Is       Active Active    Can    Copy
Name                       Complete Now    On Reboot Delete Status
-------------------------- -------- ------ --------- ------ ----------
zfsBE                      yes      no     no        yes    -
zfs2BE                     yes      yes    yes       no     -
```

Si votre système SPARC comporte plusieurs environnements d'initialisation ZFS, vous pouvez également utiliser la commande `boot -L` pour initialiser le système à partir d'un autre environnement d'initialisation que celui par défaut. Cependant, un environnement d'initialisation démarrant à partir d'une session `boot -L` n'est pas redéfini en tant qu'environnement d'initialisation par défaut et la propriété `bootfs` n'est pas mise à jour. Si vous souhaitez qu'un environnement d'initialisation démarre par défaut à partir d'une session `boot -L`, vous devez activer ce dernier avec la commande `luactivate`.

Par exemple :

```
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
```

EXEMPLE 4–12 SPARC : Initialisation d'un système de fichiers ZFS en mode de secours

Vous pouvez initialiser un système SPARC à partir de l'archive de secours située dans `/platform/‘uname -i‘/failsafe`, comme suit .

```
ok boot -F failsafe
```

EXEMPLE 4–12 SPARC : Initialisation d'un système de fichiers ZFS en mode de secours *(Suite)*

Pour initialiser une archive de secours à partir d'un jeu de données amorçable ZFS donné, employez une syntaxe du type suivant :

```
ok boot -Z rpool/ROOT/zfsBE -F failsafe
```

x86 : initialisation à partir d'un système de fichiers root ZFS

Les entrées suivantes sont ajoutées au fichier */pool-name* `/boot/grub/menu.lst` au cours du processus d'installation du SE Oracle Solaris ou de la procédure Live Upgrade pour initialiser ZFS automatiquement :

```
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
```

Si le périphérique identifié par GRUB comme périphérique d'initialisation contient un pool de stockage ZFS, le fichier `menu.lst` est utilisé pour créer le menu GRUB.

Sur un système x86 contenant plusieurs environnements d'initialisation ZFS, vous pouvez sélectionner un environnement d'initialisation à partir du menu GRUB. Si le système de fichiers root correspondant à cette entrée de menu est un jeu de données ZFS, l'option suivante est ajoutée:

```
-B $ZFS-BOOTFS
```

EXEMPLE 4–13 x86 : Initialisation d'un système de fichiers ZFS

Lors de l'initialisation d'un système à partir d'un système de fichiers ZFS, le périphérique root est spécifié par le paramètre d'initialisation `-B $ZFS-BOOTFS`. Par exemple :

```
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
```

EXEMPLE 4–14 x86 : Initialisation d'un système de fichiers ZFS en mode de secours

L'archive x86 de secours est `/boot/x86.miniroot-safe` et peut être initialisée en sélectionnant l'entrée "Solaris failsafe" dans le menu GRUB. Par exemple :

```
title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Résolution de problèmes de point de montage empêchant l'initialisation (Solaris 10 10/08)

Le meilleur moyen de changer l'environnement d'initialisation (BE) actif est d'utiliser la commande `luactivate`. En cas d'échec de l'initialisation de l'environnement d'initialisation actif, qu'il soit dû à un patch défectueux ou à une erreur de configuration, le seul moyen de procéder à l'initialisation à partir d'un autre environnement d'initialisation consiste à le sélectionner lors de l'initialisation. Vous pouvez sélectionner un autre environnement d'initialisation en effectuant une initialisation explicite à partir de la PROM sur un système SPARC ou du menu GRUB sur un système x86.

Etant donné la présence d'un bogue dans la fonctionnalité Live Upgrade de Solaris 10 10/08, il se peut que l'environnement d'initialisation inactif ne puisse pas s'initialiser. Ce problème est lié à la présence d'un point de montage non valide dans les jeux de données ZFS ou dans le jeu de données ZFS d'une zone de l'environnement d'initialisation. Le même bogue empêche également le montage de l'environnement d'initialisation s'il dispose d'un jeu de données `/var` distinct.

Si le jeu de données ZFS d'une zone possède un point de montage non valide, le point de montage peut être corrigé en procédant comme suit.

▼ Résolution des problèmes de point de montage ZFS

1. **Initialisez le système à partir d'une archive de secours.**

2. **Importez le pool.**

   Par exemple :

   ```
   # zpool import rpool
   ```

3. **Vérifiez qu'elle ne contient aucun point de montage temporaire erroné.**

   Par exemple :

   ```
   # 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
```

Le point de montage pour l'environnement d'initialisation root (`rpool/ROOT/s10up`) doit être `/`.

Si l'initialisation échoue à cause de problèmes de montage `/var`, recherchez un point de montage temporaire erroné similaire pour le jeu de données `/var`.

4 Réinitialisez les points de montage pour l'environnement d'initialisation ZFS et ses jeux de données.

Par exemple :

```
# zfs inherit -r mountpoint rpool/ROOT/s10up
# zfs set mountpoint=/ rpool/ROOT/s10up
```

5 Réinitialisez le système.

Lorsque l'option d'initialisation d'un environnement d'initialisation spécifique est proposée, à l'invite de la PROM OpenBoot ou dans le menu GRUB, sélectionnez l'environnement d'initialisation dont les points de montage viennent d'être corrigés.

Initialisation à des fins de récupération dans un environnement root ZFS

Suivez la procédure suivante si vous devez initialiser le système pour pouvoir récupérer un mot de passe `root` perdu ou résoudre tout problème similaire.

Vous devez effectuer une initialisation en mode de secours ou procéder à l'initialisation à partir d'autres médias en fonction de la gravité de l'erreur. En règle générale, vous pouvez effectuer une initialisation en mode de secours pour récupérer un mot de passe `root` perdu ou inconnu.

- "Initialisation d'un système de fichiers ZFS en mode de secours" à la page 168
- "Initialisation d'un système de fichiers ZFS à partir d'un autre média" à la page 169

Si vous devez récupérer un pool root ou un instantané de pool root, reportez-vous à la section "Restauration du pool root ZFS ou des instantanés du pool root" à la page 170.

▼ Initialisation d'un système de fichiers ZFS en mode de secours

1 Effectuez une initialisation en mode de secours.

- Sur un système SPARC, saisissez ce qui suit à l'invite `ok` :

```
ok boot -F failsafe
```

- Pour les systèmes x86, sélectionnez le mode de secours à partir du menu GRUB.

2 Lorsque vous y êtes invité, montez l'environnement d'initialisation ZFS sur le répertoire `/a` :

```
.
.
.
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 Accédez au répertoire `/a/etc`.

```
# cd /a/etc
```

4 Définissez le type TERM, le cas échéant.

```
# TERM=vt100
# export TERM
```

5 Corrigez le fichier `passwd` ou `shadow`.

```
# vi shadow
```

6 Réinitialisez le système.

```
# init 6
```

Initialisation d'un système de fichiers ZFS à partir d'un autre média

Si un problème empêche le système de s'initialiser correctement ou si tout autre problème grave se produit, vous devez procéder à l'initialisation à partir d'un serveur d'installation sur le réseau ou à partir d'un DVD d'installation d'Oracle Solaris, importer le pool root, monter l'environnement d'initialisation ZFS et tenter de résoudre le problème.

1 Initialisez le système à partir d'un DVD d'installation ou du réseau.

- SPARC : sélectionnez l'une des méthodes d'initialisation suivantes :

  ```
  ok boot cdrom -s
  ok boot net -s
  ```

  Si vous n'utilisez pas l'option `-s`, vous devez quitter le programme d'installation.
- x86 : initialisez le système à partir d'un réseau ou d'un DVD local.

2 Importez le pool root et spécifiez un autre point de montage. Par exemple :

```
# zpool import -R /a rpool
```

3 Montez l'environnement d'initialisation ZFS. Par exemple :

```
# zfs mount rpool/ROOT/zfsBE
```

4 **Accédez à l'environnement d'initialisation ZFS à partir du répertoire `/a`.**

```
# cd /a
```

5 **Réinitialisez le système.**

```
# init 6
```

Restauration du pool root ZFS ou des instantanés du pool root

Les sections suivantes décrivent comment effectuer les tâches ci-dessous :

- "Remplacement d'un disque dans le pool root ZFS" à la page 170
- "Création d'instantanés de pool root" à la page 172
- "Recréation d'un pool root ZFS et restauration d'instantanés de pool root" à la page 174
- "Restauration des instantanés d'un pool root à partir d'une initialisation de secours " à la page 176

▼ Remplacement d'un disque dans le pool root ZFS

Vous pouvez être amené à remplacer un disque dans le pool root pour les raisons suivantes :

- Le pool root est trop petit et vous souhaitez remplacer un disque plus petit par un disque plus grand.
- Un disque de pool root est défectueux. Dans un pool non redondant, si le disque est défectueux et empêche l'initialisation du système, vous devrez initialiser votre système à partir d'un autre média, par exemple un DVD ou le réseau, avant de remplacer le disque du pool root.

Dans le cadre d'une configuration de pool root mise en miroir, vous pouvez tenter de remplacer le disque sans effectuer une initialisation à partir d'un autre média. Vous pouvez remplacer un disque défaillant en utilisant la commande `zpool replace`. Si vous disposez d'un autre disque, vous pouvez également utiliser la commande `zpool attach`. Pour savoir comment connecter un autre disque et déconnecter un disque de pool root, reportez-vous à la procédure de cette section.

Avec certains composants matériels, vous devez déconnecter le disque et en supprimer la configuration avant de tenter d'utiliser la commande `zpool replace` pour remplacer le disque défectueux. Par exemple :

```
# 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
```

Avec certains composants matériels, il n'est pas nécessaire de connecter le disque de remplacement, ni de le reconfigurer après son insertion.

Vous devez identifier les noms du chemin d'accès au périphérique d'initialisation du disque actuel et du nouveau disque afin de tester l'initialisation à partir du disque de remplacement et de pouvoir initialiser le système manuellement à partir du disque existant, en cas de dysfonctionnement du disque de remplacement. Dans l'exemple de la procédure suivante, le nom du chemin du disque de pool root actuel (`c1t10d0s0`) est le suivant :

```
/pci@8,700000/pci@3/scsi@5/sd@a,0
```

Le nom du chemin du disque d'initialisation de remplacement (`c1t9d0s0`) est le suivant :

```
/pci@8,700000/pci@3/scsi@5/sd@9,0
```

1. **Connectez physiquement le disque de remplacement (nouveau disque).**

2. **Confirmez que le nouveau disque possède une étiquette SMI et une tranche 0.**

   Pour plus d'informations sur le nouvel étiquetage d'un disque destiné au pool root, reportez-vous aux références suivantes :

   - SPARC : "How to Set Up a Disk for a ZFS Root File System" du manuel *System Administration Guide: Devices and File Systems*
   - x86 : "How to Set Up a Disk for a ZFS Root File System" du manuel *System Administration Guide: Devices and File Systems*

3. **Associez le nouveau disque au pool root.**

   Par exemple :

   ```
   # zpool attach rpool c1t10d0s0 c1t9d0s0
   ```

4. **Confirmez le statut du pool root.**

   Par exemple :

   ```
   # 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 Si vous remplacez un disque de pool root par un disque plus grand, définissez la propriété de pool `autoexpand` pour augmenter la taille du pool.

Déterminez la taille du pool `rpool` existant :

```
# 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
```

Vérifiez la taille du pool `rpool` étendu :

```
# zpool list rpool
NAME   SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool  279G   146K  279G   0%  1.00x  ONLINE  -
```

6 Vérifiez que vous pouvez initialiser le système à partir du nouveau disque.

Pour les systèmes SPARC, vous devez par exemple respecter une syntaxe semblable à la suivante :

```
ok boot /pci@8,700000/pci@3/scsi@5/sd@9,0
```

7 Si le système s'initialise à partir du nouveau disque, déconnectez l'ancien disque.

Par exemple :

```
# zpool detach rpool c1t10d0s0
```

8 Configurez le système pour qu'il s'initialise automatiquement à partir du nouveau disque en réinitialisant le périphérique d'initialisation par défaut.

- SPARC : utilisez la commande `eeprom` ou `setenv` à partir de la PROM d'initialisation SPARC.
- x86 : reconfigurez le BIOS du système.

▼ Création d'instantanés de pool root

Vous pouvez créer des instantanés de pool root à des fins de récupération. La meilleure façon de créer des instantanés de pool root consiste à effectuer un instantané récursif du pool root.

La procédure suivante crée un instantané de pool root récursif et le stocke en tant que fichier et en tant qu'instantanés dans un pool sur un système distant. En cas de défaillance du pool root, le

jeu de données distant peut être monté à l'aide de NFS et le fichier d'instantané peut être reçu dans le pool recréé. Vous pouvez également stocker les instantanés de pool root en tant qu'instantanés réels dans un pool d'un système distant. L'envoi et la réception des instantanés à partir d'un système distant est un peu plus complexe, car vous devez configurer `ssh` ou utiliser `rsh` pendant que le système à réparer est initialisé à partir du miniroot du système d'exploitation Oracle Solaris.

La validation des instantanés stockés à distance en tant que fichiers ou instantanés est une étape importante dans la récupération du pool root. En appliquant l'une des deux méthodes, les instantanés doivent être recréés régulièrement, par exemple, lorsque la configuration du pool est modifiée ou lorsque le SE Solaris est mis à niveau.

Dans la procédure suivante, le système est initialisé à partir de l'environnement d'initialisation `zfsBE`.

1 Créez un pool et un système de fichiers sur un système distant pour stocker les instantanés.

Par exemple :

```
remote# zfs create rpool/snaps
```

2 Partagez le système de fichiers avec le système local.

Par exemple :

```
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 Créez un instantané récursif du pool root.

```
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 Envoyez les instantanés du pool root au système distant.

Par exemple, pour envoyer les instantanés de pool root à un pool distant sous forme de fichier, utilisez une syntaxe semblable à la suivante :

```
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
```

Pour envoyer les instantanés de pool root à un pool distant sous la forme d'instantanés, utilisez une syntaxe semblable à la suivante :

```
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
```

▼ Recréation d'un pool root ZFS et restauration d'instantanés de pool root

Dans cette procédure, on suppose les conditions suivantes :

- Le pool root ZFS ne peut pas être récupéré.
- Les instantanés du pool root ZFS sont stockés sur un système distant et sont partagés sur NFS.

Toutes les étapes sont effectuées sur le système local.

1 Initialisez le système à partir d'un DVD d'installation ou du réseau.

- SPARC : sélectionnez l'une des méthodes d'initialisation suivantes :

```
ok boot net -s
ok boot cdrom -s
```

Si vous n'utilisez pas l'option `-s`, vous devrez quitter le programme d'installation.

- x86 : initialisez le système à partir du DVD ou du réseau. Quittez ensuite le programme d'installation.

2 Montez le système de fichiers d'instantanés distant si vous avez envoyé les instantanés du pool root au système distant sous forme de fichier.

Par exemple :

```
# mount -F nfs remote-system:/rpool/snaps /mnt
```

Si vos services réseau ne sont pas configurés, il peut être nécessaire de spécifier l'adresse IP du *système distant*.

3 Si le disque du pool root est remplacé et ne contient aucune étiquette de disque pouvant être utilisée par ZFS, vous devez renommer le disque.

Pour plus d'informations sur le nouvel étiquetage du disque, reportez-vous aux références suivantes :

- SPARC : "How to Set Up a Disk for a ZFS Root File System" du manuel *System Administration Guide: Devices and File Systems*
- x86 : "How to Set Up a Disk for a ZFS Root File System" du manuel *System Administration Guide: Devices and File Systems*

4 Recréez le pool root.

Par exemple :

```
# zpool create -f -o failmode=continue -R /a -m legacy -o cachefile=
/etc/zfs/zpool.cache rpool c1t1d0s0
```

5 Restaurez les instantanés du pool root.

Cette étape peut prendre un certain temps. Par exemple :

```
# cat /mnt/rpool.snap1 | zfs receive -Fdu rpool
```

L'utilisation de l'option `-u` implique que l'archive restaurée n'est pas montée à la fin de l'opération `zfs receive`.

Pour restaurer les instantanés d'un pool root stockés dans un pool sur un système distant, utilisez une syntaxe semblable à la suivante :

```
# ssh remote-system zfs send -Rb tank/snaps/rpool@snap1 | zfs receive -F rpool
```

6 Vérifiez que les jeux de données du pool root sont restaurés.

Par exemple :

```
# zfs list
```

7 Définissez la propriété `bootfs` sur l'environnement d'initialisation du pool root.

Par exemple :

```
# zpool set bootfs=rpool/ROOT/zfsBE rpool
```

8 Installez les blocs d'initialisation sur le nouveau disque.

- 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 Réinitialisez le système.

```
# init 6
```

▼ Restauration des instantanés d'un pool root à partir d'une initialisation de secours

Cette procédure part du principe que les instantanés du pool root existant sont disponibles. Dans l'exemple suivant, ils se trouvent sur le système local.

```
# 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 Arrêtez le système et initialisez-le en mode secours.

```
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 Annulez (roll back) chaque instantané du pool root.

```
# zfs rollback rpool@snap1
# zfs rollback rpool/ROOT@snap1
# zfs rollback rpool/ROOT/s10zfsBE@snap1
```

3 Réinitialisez le système en mode multiutilisateur.

```
# init 6
```

◆ ◆ ◆ CHAPITRE 5

Gestion des systèmes de fichiers Oracle Solaris ZFS

Ce chapitre contient des informations détaillées sur la gestion des systèmes de fichiers ZFS Oracle Solaris. Il aborde notamment les concepts d'organisation hiérarchique des systèmes de fichiers, d'héritage des propriétés, de gestion automatique des points de montage et d'interaction sur les partages.

Ce chapitre contient les sections suivantes :

- "Gestion des systèmes de fichiers ZFS (présentation)" à la page 179
- "Création, destruction et renommage de systèmes de fichiers ZFS" à la page 180
- "Présentation des propriétés ZFS" à la page 183
- "Envoi de requêtes sur les informations des systèmes de fichiers ZFS" à la page 198
- "Gestion des propriétés ZFS" à la page 200
- "Montage de système de fichiers ZFS" à la page 205
- "Activation et annulation du partage des systèmes de fichiers ZFS" à la page 210
- "Définition des quotas et réservations ZFS" à la page 212
- "Mise à niveau des systèmes de fichiers ZFS" à la page 218

Gestion des systèmes de fichiers ZFS (présentation)

La création d'un système de fichiers ZFS s'effectue sur un pool de stockage. La création et la destruction des systèmes de fichiers peuvent s'effectuer de manière dynamique, sans allocation ni formatage manuels de l'espace disque sous-jacent. En raison de leur légèreté et de leur rôle central dans l'administration du système ZFS, la création de ces systèmes de fichiers constitue généralement une opération extrêmement courante.

La gestion des systèmes de fichiers ZFS s'effectue à l'aide de la commande `zfs`. La commande `zfs` offre un ensemble de sous-commandes permettant d'effectuer des opérations spécifiques sur les systèmes de fichiers. Chacune de ces sous-commandes est décrite en détail dans ce chapitre. Cette commande permet également de gérer les instantanés, les volumes et les clones. Toutefois, ces fonctionnalités sont uniquement traitées de manière succincte dans ce chapitre. Pour plus d'informations sur les instantanés et les clones, reportez-vous au Chapitre 6,

"Utilisation des instantanés et des clones ZFS Oracle Solaris". Pour de plus amples informations sur les volumes ZFS, reportez-vous à la section "Volumes ZFS" à la page 281.

Remarque – Dans ce chapitre, le terme *jeu de données* désigne de manière générique un système de fichiers, un instantané, un clone ou un volume.

Création, destruction et renommage de systèmes de fichiers ZFS

La création et la destruction des systèmes de fichiers ZFS s'effectuent respectivement à l'aide des commandes `zfs create` et `zfs destroy`. Vous pouvez renommer les systèmes de fichiers ZFS en utilisant la commande `zfs rename`.

- "Création d'un système de fichiers ZFS" à la page 180
- "Destruction d'un système de fichiers ZFS" à la page 181
- "Modification du nom d'un système de fichiers ZFS" à la page 182

Création d'un système de fichiers ZFS

La création des systèmes de fichiers ZFS s'effectue à l'aide de la commande `zfs create`. La sous-commande `create` ne peut contenir qu'un argument : le nom du système de fichiers à créer. Le nom de ce système de fichiers permet également de définir le nom du chemin par rapport au nom du pool, comme suit :

pool-name/[filesystem-name/]filesystem-name

Le nom du pool et les noms des systèmes de fichiers existants mentionnés dans le chemin déterminent l'emplacement du nouveau système de fichiers dans la structure hiérarchique. Le dernier nom mentionné dans le chemin correspond au nom du système de fichiers à créer. Ce nom doit respecter les conventions d'attribution de nom définies à la section "Exigences d'attribution de noms de composants ZFS" à la page 31.

Dans l'exemple suivant, un système de fichiers nommé `jeff` est créé dans le système de fichiers `tank/home`.

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

Si le processus de création se déroule correctement, le système de fichiers ZFS est automatiquement monté. Par défaut, les systèmes de fichiers sont montés sous */dataset*, à l'aide du chemin défini pour le nom du système dans la commande `create`. Dans cet exemple, le système de fichiers `jeff` créé est monté sous `/tank/home/jeff`. Pour plus d'informations sur les points de montage gérés automatiquement, reportez-vous à la section "Gestion des points de montage ZFS" à la page 206.

Pour plus d'informations sur la commande `zfs create`, reportez-vous à la page de manuel `zfs(1M)`.

Il est possible de définir les propriétés du système de fichiers lors de la création de ce dernier.

Dans l'exemple ci-dessous, le point de montage `/export/zfs` est créé pour le système de fichiers `tank/home` :

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

Pour plus d'informations sur les propriétés des systèmes de fichiers, reportez-vous à la section "Présentation des propriétés ZFS" à la page 183.

Destruction d'un système de fichiers ZFS

La destruction d'un système de fichiers ZFS s'effectue à l'aide de la commande `zfs destroy`. Les systèmes de fichiers détruits sont automatiquement démontés et ne sont plus partagés. Pour plus d'informations sur les montages ou partages gérés automatiquement, reportez-vous à la section "Points de montage automatiques" à la page 206.

L'exemple suivant illustre la destruction du système de fichiers `tank/home/mark` :

```
# zfs destroy tank/home/mark
```

Attention – Aucune invite de confirmation ne s'affiche lors de l'exécution de la sous-commande `destroy`. Son utilisation requiert une attention particulière.

Si le système de fichiers à détruire est occupé et ne peut pas être démonté, la commande `zfs destroy` échoue. Pour détruire un système de fichiers actif, indiquez l'option `-f`. L'utilisation de cette option requiert une attention particulière. En effet, elle permet de démonter, d'annuler le partage et de détruire des systèmes de fichiers actifs, ce qui risque d'affecter le comportement de certaines applications.

```
# zfs destroy tank/home/matt
cannot unmount 'tank/home/matt': Device busy

# zfs destroy -f tank/home/matt
```

La commande `zfs destroy` échoue également si le système de fichiers possède des descendants. Pour détruire un système de fichiers et l'ensemble des descendants de ce système de fichiers, indiquez l'option `-r`. Ce type d'opération de destruction récursive entraîne également la destruction des instantanés ; l'utilisation de cette option requiert donc une attention particulière.

```
# 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
```

Si le système de fichiers à détruire possède des systèmes indirectement dépendants, même la commande de destruction récursive échoue. Pour forcer la destruction de *tous* les systèmes dépendants, y compris des systèmes de fichiers clonés situés en dehors de la structure hiérarchique cible, vous devez indiquer l'option `-R`. Utilisez cette option avec précaution.

```
# 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
```

Attention – Aucune invite de confirmation ne s'affiche lors de l'utilisation des options `-f`, `-r` ou `-R` avec la commande `zfs destroy`. L'utilisation de ces options requiert donc une attention particulière.

Pour plus d'informations sur les instantanés et les clones, reportez-vous au Chapitre 6, "Utilisation des instantanés et des clones ZFS Oracle Solaris".

Modification du nom d'un système de fichiers ZFS

La modification du nom d'un système de fichiers ZFS s'effectue à l'aide de la commande `zfs rename`. La commande `rename` permet d'effectuer les opérations suivantes :

- Modifier le nom d'un système de fichiers.
- Déplacer le système de fichiers au sein de la hiérarchie ZFS.
- Modifier le nom d'un système de fichiers et son emplacement au sein de la hiérarchie ZFS.

L'exemple suivant utilise la sous-commande `rename` pour renommer un système de fichiers `eric` en `eric_old` :

```
# zfs rename tank/home/eric tank/home/eric_old
```

L'exemple ci-dessous illustre la modification de l'emplacement d'un système de fichiers à l'aide de la sous-commande `zfs rename` :

```
# zfs rename tank/home/mark tank/ws/mark
```

Dans cet exemple, le système de fichiers `mark` est déplacé de `tank/home` vers `tank/ws`. Lorsque vous modifiez l'emplacement d'un système de fichiers à l'aide de la commande rename, le nouvel emplacement doit se trouver au sein du même pool et l'espace disque disponible doit être suffisant pour contenir le nouveau système de fichiers. Lorsque le nouvel emplacement ne dispose pas de suffisamment d'espace disque, l'opération `rename` échoue.

Pour plus d'informations sur les quotas, reportez-vous à la section “Définition des quotas et réservations ZFS” à la page 212.

L'opération `rename` tente de démonter, puis de remonter le système de fichiers ainsi que ses éventuels systèmes de fichiers descendants. Si la commande `rename` ne parvient pas à démonter un système de fichiers actif, l'opération échoue. Si ce problème survient, vous devez forcer le démontage du système de fichiers.

Pour plus d'informations sur la modification du nom des instantanés, reportez-vous à la section “Renommage d'instantanés ZFS” à la page 222.

Présentation des propriétés ZFS

Les propriétés constituent le mécanisme principal de contrôle du comportement des systèmes de fichiers, des volumes, des instantanés et des clones. Sauf mention contraire, les propriétés définies dans cette section s'appliquent à tous les types de jeux de données.

- “Propriétés ZFS natives en lecture seule” à la page 192
- “Propriétés ZFS natives définies” à la page 193
- “Propriétés ZFS définies par l'utilisateur” à la page 196

Les propriétés se divisent en deux catégories : les propriétés natives et les propriétés définies par l'utilisateur. Les propriétés natives permettent de fournir des statistiques internes ou de contrôler le comportement du système de fichiers ZFS. Certaines de ces propriétés peuvent être définies tandis que d'autres sont en lecture seule. Les propriétés définies par l'utilisateur n'ont aucune incidence sur le comportement des systèmes de fichiers ZFS. En revanche, elles permettent d'annoter les jeux de données avec des informations adaptées à votre environnement. Pour plus d'informations sur les propriétés définies par l'utilisateur, reportez-vous à la section “Propriétés ZFS définies par l'utilisateur” à la page 196.

La plupart des propriétés pouvant être définies peuvent également être héritées. Les propriétés héritables sont des propriétés qui, une fois définies sur un système de fichiers parent, s'appliquent à l'ensemble de ses descendants.

Toutes ces propriétés héritables sont associées à une source indiquant la façon dont la propriété a été obtenue. Les sources de propriétés peuvent être définies sur les valeurs suivantes :

`local`	Indique que la propriété été définie de manière explicite sur le jeu de données à l'aide de la commande `zfs set`, selon la procédure décrite à la section “Définition des propriétés ZFS” à la page 200.
`inherited from` *dataset-name*	Indique que la propriété a été héritée à partir de l'ascendant indiqué.
`default`	Indique que la valeur de la propriété n'a été ni héritée, ni définie en local. Cette source est définie lorsque la propriété

n'est pas définie en tant que source `local` sur aucun système ascendant.

Le tableau suivant répertorie les propriétés de système de fichiers ZFS natives en lecture seule et pouvant être définies. Les propriétés natives en lecture seule sont signalées comme tel. Les autres propriétés natives répertoriées dans le tableau peuvent être définies. Pour plus d'informations sur les propriétés définies par l'utilisateur, reportez-vous à la section "Propriétés ZFS définies par l'utilisateur" à la page 196.

TABLEAU 5–1 Description des propriétés ZFS natives

Nom de propriété	Type	Valeur par défaut	Description
`aclinherit`	Chaîne	`secure`	Contrôle le processus d'héritage des entrées ACL lors de la création de fichiers et de répertoires. Les valeurs possibles sont `discard`, `noallow`, `secure` et `passthrough`. Pour une description de ces valeurs, reportez-vous à la section "Propriétés ACL" à la page 247.
`aclmode`	Chaîne	`groupmask`	Contrôle le processus de modification des entrées ACL lors des opérations `chmod`. Les valeurs possibles sont `discard`, `groupmask` et `passthrough`. Pour une description de ces valeurs, reportez-vous à la section "Propriétés ACL" à la page 247.
`atime`	Booléen	`on`	Détermine si l'heure d'accès aux fichiers est mise à jour lorsqu'ils sont consultés. La désactivation de cette propriété évite de produire du trafic d'écriture lors de la lecture de fichiers et permet parfois d'améliorer considérablement les performances ; elle risque cependant de perturber les logiciels de messagerie et autres utilitaires du même type.
`available`	Valeur numérique	SO	Propriété en lecture seule indiquant la quantité d'espace disque disponible pour un système de fichiers et l'ensemble de ses enfants, sans tenir compte des autres activités du pool. L'espace disque étant partagé au sein d'un pool, l'espace disponible peut être limité par divers facteurs, y compris la taille du pool physique, les quotas, les réservations ou les autres jeux de données présents au sein du pool. L'abréviation de la propriété est `avail`. Pour plus d'informations sur la détermination de l'espace disque, reportez-vous à la section "Comptabilisation de l'espace disque ZFS" à la page 32.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`canmount`	Booléen	`on`	Détermine si un système de fichiers donné peut être monté à l'aide de la commande `zfs mount`. Cette propriété peut être définie sur tous les systèmes de fichiers et ne peut pas être héritée. En revanche, lorsque cette propriété est définie sur `off`, un point de montage peut être hérité par des systèmes de fichiers descendants. Le système de fichiers à proprement parler n'est toutefois pas monté. Pour plus d'informations, reportez-vous à la section "Propriété `canmount`" à la page 195.
`checksum`	Chaîne	`on`	Détermine la somme de contrôle permettant de vérifier l'intégrité des données. La valeur par défaut est définie sur `on`. Cette valeur permet de sélectionner automatiquement l'algorithme approprié, actuellement `fletcher4`. Les valeurs possibles sont `on`, `off`, `fletcher2`, `fletcher4` et `sha256`. La valeur `off` entraîne la désactivation du contrôle d'intégrité des données utilisateur. La valeur `off` n'est pas recommandée.
`compression`	Chaîne	`off`	Active ou désactive la compression d'un jeu de données. Les valeurs sont `on` , `off` et `lzjb`, `gzip` ou `gzip-`*N*. Donner à cette propriété la valeur `lzjb`, `gzip` ou la valeur `gzip-` *N* a actuellement le même effet que la valeur `on`. L'activation de la compression sur un système de fichiers contenant des données existantes entraîne uniquement la compression des nouvelles données. Les données actuelles restent non compressées. L'abréviation de la propriété est `compress`.
`compressratio`	Valeur numérique	SO	Propriété en lecture seule indiquant le ratio de compression obtenu pour un jeu de données, exprimé sous la forme d'un multiple. La compression peut être activée en exécutant la commande `zfs set compression=on` *dataset*. Cette valeur est calculée sur la base de la taille logique de l'ensemble des fichiers et de la quantité de données physiques indiquée. Elle induit un gain explicite basé sur l'utilisation de la propriété `compression`.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`copies`	Valeur numérique	`1`	Définit le nombre de copies des données utilisateur par système de fichiers. Les valeurs disponibles sont `1`, `2` ou `3`. Ces copies viennent s'ajouter à toute redondance au niveau du pool. L'espace disque utilisé par plusieurs copies de données utilisateur est chargé dans le fichier et le jeu de données correspondants et pénalise les quotas et les réservations. En outre, la propriété `used` est mise à jour lorsque plusieurs copies sont activées. Considérez la définition de cette propriété à la création du système de fichiers car lorsque vous la modifiez sur un système de fichiers existant, les modifications ne s'appliquent qu'aux nouvelles données.
`creation`	Chaîne	SO	Propriété en lecture seule identifiant la date et l'heure de création d'un jeu de données.
`devices`	Booléen	`on`	Contrôle si les fichiers de périphérique d'un système de fichiers peuvent être ouverts.
`exec`	Booléen	`on`	Contrôle l'autorisation d'exécuter les programmes dans un système de fichiers. Par ailleurs, lorsqu'elle est définie sur `off`, les appels de la commande `mmap(2)` avec `PROT_EXEC` ne sont pas autorisés.
`mounted`	Booléen	SO	Propriété en lecture seule indiquant si un système de fichiers, un clone ou un instantané est actuellement monté. Cette propriété ne s'applique pas aux volumes. Les valeurs possibles sont `yes` ou `no`.
`mountpoint`	Chaîne	SO	Détermine le point de montage utilisé pour le système de fichiers. Lorsque la propriété `mountpoint` d'un système de fichiers est modifiée, ce système de fichiers ainsi que les éventuels systèmes descendants héritant du point de montage sont démontés. Si la nouvelle valeur est définie sur `legacy`, ces systèmes restent démontés. Dans le cas contraire, ils sont automatiquement remontés au nouvel emplacement si la propriété était précédemment définie sur `legacy` ou sur `none` ou s'ils étaient montés avant la modification de la propriété. D'autre part, le partage de tout système de fichiers est annulé puis rétabli au nouvel emplacement. Pour plus d'informations sur l'utilisation de cette propriété, reportez-vous à la section "Gestion des points de montage ZFS" à la page 206.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`primarycache`	Chaîne	`all`	Contrôle les éléments mis en cache dans le cache principal (ARC). Les valeurs possibles sont `all`, `none` et `metadata`. Si elles sont définies sur `all`, les données d'utilisateur et les métadonnées sont mises en cache. Si elle est définie sur `none`, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si elles sont définies sur `metadata`, seules les métadonnées sont mises en cache. Lorsque ces propriétés sont définies sur des systèmes de fichiers existants, seule la nouvelle E/S est mise en cache en fonction de la valeur de ces propriétés. Certains environnements de bases de données pourraient bénéficier de la non-mise en cache des données d'utilisateur. Vous devez déterminer si la configuration des paramètres du cache est adaptée à votre environnement.
`origin`	Chaîne	SO	Propriété en lecture seule appliquée aux systèmes de fichiers ou aux volumes clonés et indiquant l'instantané à partir duquel le clone a été créé. Le système d'origine ne peut pas être détruit (même à l'aide des options `-r` ou `-f`) tant que le clone existe. Les systèmes de fichiers non clonés indiquent un système d'origine `none`.
`quota`	Valeur numérique (ou `none`)	`none`	Limite la quantité d'espace disque qu'un système de fichiers et ses descendants peuvent consommer. Cette propriété permet d'appliquer une limite fixe à la quantité d'espace disque utilisée, y compris l'espace utilisé par les descendants, qu'il s'agisse de systèmes de fichiers ou d'instantanés. La définition d'un quota sur un descendant de système de fichiers déjà associé à un quota n'entraîne pas le remplacement du quota du système ascendant. Cette opération entraîne au contraire l'application d'une limite supplémentaire. Les quotas ne peuvent pas être définis pour les volumes car la propriété `volsize` sert de quota implicite. Pour plus d'informations concernant la définition de quotas, reportez-vous à la section "Définitions de quotas sur les systèmes de fichiers ZFS" à la page 213.
`readonly`	Booléen	`off`	Contrôle l'autorisation de modifier un jeu de données. Lorsqu'elle est définie sur `on`, aucune modification ne peut être apportée. L'abréviation de la propriété est `rdonly`.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`recordsize`	Valeur numérique	`128K`	Spécifie une taille de bloc suggérée pour les fichiers d'un système de fichiers. L'abréviation de la propriété est `recsize`. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section "Propriété `recordsize`" à la page 195.
`referenced`	Valeur numérique	SO	Propriété en lecture seule identifiant la quantité de données à laquelle un jeu de données a accès, lesquelles peuvent ou non être partagées avec d'autres jeux de données du pool. Lorsqu'un instantané ou un clone est créé, il indique dans un premier temps la même quantité d'espace disque que le système de fichiers ou l'instantané à partir duquel il a été créé. En effet, son contenu est identique. L'abréviation de la propriété est `refer`.
`refquota`	Valeur numérique (ou none)	`none`	Définit la quantité d'espace disque pouvant être utilisé par un jeu de données. Cette propriété définit une quantité d'espace maximale. Cette limite fixe n'inclut pas l'espace disque utilisé par les descendants, tels que les instantanés et les clones.
`refreservation`	Valeur numérique (ou none)	`none`	Définit la quantité d'espace disque minimale garantie pour un jeu de données, à l'exclusion des descendants, notamment les instantanés et les clones. Lorsque la quantité d'espace disque utilisée est inférieure à cette valeur, le système considère que le jeu de donnée utilise la quantité d'espace spécifiée par `refreservation` La réservation `refreservation` est prise en compte dans l'espace disque utilisé des jeux de donnés parent et vient en déduction de leurs quotas et réservations. Lorsque la propriété `refreservation` est définie, un instantané n'est autorisé que si suffisamment d'espace est disponible dans le pool au-delà de cette réservation afin de pouvoir contenir le nombre actuel d'octets *référencés* dans le jeu de données. L'abréviation de la propriété est `refserv`.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`reservation`	Valeur numérique (ou none)	`none`	Définit la quantité d'espace disque minimale garantie à un système de fichiers et ses descendants. Lorsque la quantité d'espace disque utilisée est inférieure à cette valeur, le système considère que le système de fichiers utilise la quantité d'espace réservée. Les réservations sont prises en compte dans l'espace disque utilisé du système de fichiers parent, et viennent en déduction des quotas et réservations de celui-ci. L'abréviation de la propriété est `reserv`. Pour plus d'informations, reportez-vous à la section "Définition de réservations sur les systèmes de fichiers ZFS" à la page 216.
`secondarycache`	Chaîne	`all`	Contrôle les éléments qui sont mis en cache dans le cache secondaire (L2ARC). Les valeurs possibles sont `all`, `none` et `metadata`. Si elles sont définies sur `all`, les données d'utilisateur et les métadonnées sont mises en cache. Si elle est définie sur `none`, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si elles sont définies sur `metadata`, seules les métadonnées sont mises en cache.
`setuid`	Booléen	`on`	Contrôle l'application du bit `setuid` dans un système de fichiers.
`shareiscsi`	Chaîne	off	Contrôle si un volume ZFS est partagé en tant que cible iSCSI. Les valeurs de la propriété sont `on`, `off` et `type=disk`. Vous pouvez définir la valeur `shareiscsi=on` pour un système de fichiers afin que tous les volumes ZFS au sein du système de fichiers soient partagés par défaut. Cependant, la configuration de cette propriété sur un système de fichiers n'a aucune incidence directe.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`sharenfs`	Chaîne	`off`	Détermine si un système de fichiers est disponible via NFS, ainsi que les options utilisées. Si cette propriété est définie sur `on`, la commande `zfs share` est exécutée sans option. Dans le cas contraire, la commande `zfs share` est exécutée avec les options équivalentes au contenu de cette propriété. Si elle est définie sur `off`, le système de fichiers est géré par l'intermédiaire des commandes `share` et `unshare` ainsi que par le fichier `dfstab` existants. Pour plus d'informations sur le partage des systèmes de fichiers ZFS, reportez-vous à la section "Activation et annulation du partage des systèmes de fichiers ZFS" à la page 210.
`snapdir`	Chaîne	`hidden`	Détermine si le répertoire `.zfs` doit être affiché ou masqué au niveau du root du système de fichiers. Pour plus d'informations sur l'utilisation des instantanés, reportez-vous à la section "Présentation des instantanés ZFS" à la page 219.
`type`	Chaîne	SO	Propriété en lecture seule identifiant le type de jeu de données comme étant un système de fichiers (`filesystem` ; système de fichiers à proprement parler ou clone), un volume (`volume`) ou un instantané (`snapshot`).
`used`	Valeur numérique	SO	Propriété en lecture seule identifiant la quantité d'espace disque utilisée par le jeu de données et tous ses descendants. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section "Propriété `used`" à la page 193.
`usedbychildren`	Valeur numérique	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par les enfants de ce jeu de données, qui serait libérée si tous ses enfants étaient détruits. L'abréviation de la propriété est `usedchild`.
`usedbydataset`	Valeur numérique	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par le jeu de données lui-même, qui serait libérée si ce dernier était détruit, après la destruction préalable de tous les instantanés et la suppression de toutes les réservations `refreservation`. L'abréviation de la propriété est `usedds`.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`usedbyrefreservation`	Valeur numérique	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par un jeu `refreservation` sur un jeu de données, qui serait libérée si le jeu `refreservation` était supprimé. L'abréviation de la propriété est `usedrefreserv`.
`usedbysnapshots`	Valeur numérique	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par les instantanés de ce jeu de données. En particulier, elle correspond à la quantité d'espace disque qui serait libérée si l'ensemble des instantanés de ce jeu de données était supprimé. Notez que cette valeur ne correspond pas simplement à la somme des propriétés `used` des instantanés, car l'espace peut être partagé par plusieurs instantanés. L'abréviation de la propriété est `usedsnap`.
`version`	Valeur numérique	SO	Identifie la version du disque d'un système de fichiers. Cette information n'est pas liée à la version du pool. Cette propriété peut uniquement être définie avec une version supérieure prise en charge par la version du logiciel. Pour plus d'informations, reportez-vous à la commande `zfs upgrade`.
`volsize`	Valeur numérique	SO	Spécifie la taille logique des volumes. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section "Propriété `volsize`" à la page 196.
`volblocksize`	Valeur numérique	`8 KB`	(Volumes) Spécifie la taille de bloc du volume. Une fois que des données ont été écrites sur un volume, la taille de bloc ne peut plus être modifiée. Vous devez donc définir cette valeur lors de la création du volume. La taille de bloc par défaut des volumes est de 8 Ko. Toute puissance de deux comprise entre 512 octets et 128 Ko est correcte. L'abréviation de la propriété est `volblock`.

TABLEAU 5–1 Description des propriétés ZFS natives *(Suite)*

Nom de propriété	Type	Valeur par défaut	Description
`zoned`	Booléen	SO	Indique si un système de fichiers a été ajouté ou non à une zone globale. Si cette propriété est activée, le point de montage ne figure pas dans la zone globale et le système ZFS ne peut pas monter le système de fichiers en réponse aux requêtes. Lors de la première installation d'une zone, cette propriété est définie pour tous les systèmes de fichiers ajoutés. Pour plus d'informations sur l'utilisation du système ZFS avec des zones installées, reportez-vous à la section "Utilisation de ZFS dans un système Solaris avec zones installées" à la page 284.
`xattr`	Booléen	`on`	Indique si les attributs étendus sont activés (`on`) ou désactivés (`off`) pour ce système de fichiers.

Propriétés ZFS natives en lecture seule

Les propriétés natives en lecture seule peuvent être récupérées, mais ne peuvent pas être définies. Elles ne peuvent pas non plus être héritées. Certaines propriétés natives sont spécifiques à un type de jeu de données. Dans ce cas, le type de jeu de données est mentionné dans la description figurant dans le Tableau 5–1.

Les propriétés natives en lecture seule sont répertoriées dans cette section et décrites dans le Tableau 5–1.

- `available`
- `compressratio`
- `creation`
- `mounted`
- `origin`
- `referenced`
- `type`
- `used`

  Pour plus d'informations sur cette propriété, reportez-vous à la section "Propriété `used`" à la page 193.
- `usedbychildren`
- `usedbydataset`
- `usedbyrefreservation`
- `usedbysnapshots`

Pour plus d'informations sur la détermination de l'espace disque, notamment sur les propriétés `used`, `referenced` et `available`, reportez-vous à la section "Comptabilisation de l'espace disque ZFS" à la page 32.

Propriété `used`

La propriété `used` est une propriété en lecture seule indiquant la quantité d'espace disque utilisée par le jeu de données et l'ensemble de ses descendants. Cette valeur est comparée au quota et à la réservation définis pour le jeu de données. La quantité d'espace disque utilisé n'inclut pas la réservation du jeu de données. En revanche, elle prend en compte les réservations définies pour les éventuels jeux de données descendants. La quantité d'espace disque utilisée sur le parent par un jeu de données, ainsi que la quantité d'espace disque libérée si le jeu de données est détruit de façon récursive, constituent la plus grande partie de son espace utilisé et sa réservation.

Lors de la création d'un instantané, l'espace disque correspondant est dans un premier temps partagé entre cet instantané et le système de fichiers ainsi que les instantanés existants (le cas échéant). Lorsque le système de fichiers est modifié, l'espace disque précédemment partagé devient dédié à l'instantané. Il est alors comptabilisé dans l'espace utilisé par cet instantané. L'espace disque utilisé par un instantané représente ses données uniques. La suppression d'instantanés peut également augmenter l'espace disque dédié et utilisé par les autres instantanés. Pour plus d'informations sur les instantanés et les questions d'espace, reportez-vous à la section "Comportement d'espace saturé" à la page 34.

La quantité d'espace disque utilisé, disponible et référencé ne comprend pas les modifications en attente. Ces modifications sont généralement prises en compte au bout de quelques secondes. La modification d'un disque utilisant la fonction `fsync(3c)` ou `O_SYNC` ne garantit pas la mise à jour immédiate des informations concernant l'utilisation de l'espace disque.

Les informations de propriété `usedbychildren`, `usedbydataset` , `usedbyrefreservation` et `usedbysnapshots` peuvent être affichées à l'aide de la commande `zfs list -o space`. Ces propriétés divisent la propriété `used` en espace disque utilisé par les descendants. Pour plus d'informations, reportez-vous au Tableau 5–1.

Propriétés ZFS natives définies

Les propriétés natives définies sont les propriétés dont les valeurs peuvent être récupérées et modifiées. La définition des propriétés natives s'effectue à l'aide de la commande `zfs set`, selon la procédure décrite à la section "Définition des propriétés ZFS" à la page 200 ou à l'aide de la commande `zfs create`, selon la procédure décrite à la section "Création d'un système de fichiers ZFS" à la page 180. A l'exception des quotas et des réservations, les propriétés natives définies sont héritées. Pour plus d'informations sur les quotas et les réservations, reportez-vous à la section "Définition des quotas et réservations ZFS" à la page 212.

Certaines propriétés natives définies sont spécifiques à un type de jeu de données. Dans ce cas, le type de jeu de données est mentionné dans la description figurant dans le Tableau 5–1. Sauf indication contraire, les propriétés s'appliquent à tous les types de jeu de données : aux systèmes de fichiers, aux volumes, aux clones et aux instantanés.

Les propriétés pouvant être définies sont répertoriées dans cette section et décrites dans le Tableau 5–1.

- `aclinherit`

  Pour obtenir une description détaillée, reportez-vous à la section "Propriétés ACL" à la page 247.
- `aclmode`

  Pour obtenir une description détaillée, reportez-vous à la section "Propriétés ACL" à la page 247.
- `atime`
- `canmount`
- `checksum`
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`

  Pour obtenir une description détaillée de cette propriété, reportez-vous à la section "Propriété `recordsize`" à la page 195.
- `refquota`
- `refreservation`
- `reservation`
- `secondarycache`
- `shareiscsi`
- `setuid`
- `snapdir`
- `version`
- `volsize`

Pour obtenir une description détaillée de cette propriété, reportez-vous à la section "Propriété volsize" à la page 196.

- volblocksize
- zoned
- xattr

Propriété canmount

Si la propriété canmount est désactivée (valeur off), le système de fichiers ne peut pas être monté à l'aide de la commande zfs mount, ni de la commande zfs mount -a. Définir cette propriété sur off équivaut à définir la propriété mountpoint sur none, à la différence près que le système de fichiers conserve une propriété mountpoint normale pouvant être héritée. Vous pouvez par exemple définir cette propriété sur la valeur off et définir des propriétés héritées pour les systèmes de fichiers descendants. Toutefois, le système de fichiers parent à proprement parler n'est jamais monté, ni accessible par les utilisateurs. Dans ce cas, le système de fichiers parent sert de *conteneur* afin de pouvoir définir des propriétés sur le conteneur ; toutefois, le conteneur à proprement parler n'est jamais accessible.

L'exemple suivant illustre la création du système de fichiers userpool avec la propriété canmount désactivée (valeur off). Les points de montage des systèmes de fichiers utilisateur descendants sont définis sur un emplacement commun, /export/home. Les systèmes de fichiers descendants héritent des propriétés définies sur le système de fichiers parent, mais celui-ci n'est jamais monté.

```
# 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
```

Propriété recordsize

La propriété recordsize spécifie une taille de bloc suggérée pour les fichiers du système de fichiers.

Cette propriété s'utilise uniquement pour les charges de travail de base de données accédant à des fichiers résidant dans des enregistrements à taille fixe. Le système ZFS ajuste automatiquement les tailles en fonction d'algorithmes internes optimisés pour les schémas d'accès classiques. Pour les bases de données générant des fichiers volumineux mais accédant uniquement à certains fragments de manière aléatoire, ces algorithmes peuvent se révéler inadaptés. La définition d'une valeur recordsize supérieure ou égale à la taille d'enregistrement de la base de données peut améliorer les performances du système de manière significative. Il est vivement déconseillé d'utiliser cette propriété pour les systèmes de fichiers à usage générique.

En outre, elle peut affecter les performances du système. La taille spécifiée doit être une puissance de 2 supérieure ou égale à 512 octets et inférieure ou égale à 128 Ko. La modification de la valeur `recordsize` du système de fichiers affecte uniquement les fichiers créés ultérieurement. Cette modification n'affecte pas les fichiers existants.

L'abréviation de la propriété est `recsize`.

Propriété `volsize`

La propriété `volsize` spécifie la taille logique du volume. Par défaut, la création d'un volume définit une réservation de taille identique. Toute modification apportée à la valeur de la propriété `volsize` se répercute dans des proportions identiques au niveau de la réservation. Ce fonctionnement permet d'éviter les comportements inattendus lors de l'utilisation des volumes. L'utilisation de volumes contenant moins d'espace disponible que la valeur indiquée risque, suivant le cas, d'entraîner des comportements non valides et des altérations de données. Ces symptômes peuvent également survenir lors de la modification et notamment de la réduction de la taille du volume en cours d'utilisation. Faites preuve de prudence lorsque vous ajustez la taille d'un volume.

Même s'il s'agit d'une opération déconseillée, vous avez la possibilité de créer des volumes fragmentés. Pour ce faire, spécifiez l'étiquette `-s` dans la commande `zfs create -V` ou modifiez la réservation, une fois le volume créé. Un *volume fragmenté* désigne un volume dont la réservation est différente de la taille de volume. Les modifications apportées à la propriété `volsize` des volumes fragmentés ne sont pas répercutées au niveau de la réservation.

Pour plus d'informations sur l'utilisation des volumes, reportez-vous à la section “Volumes ZFS” à la page 281.

Propriétés ZFS définies par l'utilisateur

Outre les propriétés natives, le système ZFS prend en charge des propriétés définies par l'utilisateur. Les propriétés définies par l'utilisateur n'ont aucune incidence sur le comportement du système ZFS. En revanche, elles permettent d'annoter les jeux de données avec des informations adaptées à votre environnement.

Les noms de propriétés définies par l'utilisateur doivent respecter les conventions suivantes :

- Elles doivent contenir le caractère ":" (deux points) afin de les distinguer des propriétés natives.
- Elles doivent contenir des lettres en minuscule, des chiffres ou les signes de ponctuation suivants : ':', '+', '.', '_'.
- La longueur maximale du nom d'une propriété définie par l'utilisateur est de 256 caractères.

La syntaxe attendue des noms de propriétés consiste à regrouper les deux composants suivants (cet espace de noms n'est toutefois pas appliqué par les systèmes ZFS) :

module:property

Si vous utilisez des propriétés définies par l'utilisateur dans un contexte de programmation, spécifiez un nom de domaine DNS inversé pour le composant *module* des noms de propriétés, afin de réduire la probabilité que deux packages développés séparément n'utilisent un nom de propriété identique à des fins différentes. Les noms de propriété commençant par `com.oracle.` sont réservés à l'usage d'Oracle Corporation.

Les valeurs des propriétés définies par l'utilisateur doivent respecter les conventions suivantes :

- Elles doivent être constituées de chaînes arbitraires systématiquement héritées et elle ne doivent jamais être validées.
- La longueur maximale de la valeur d'une propriété définie par l'utilisateur est de 1024 caractères.

Par exemple :

```
# zfs set dept:users=finance userpool/user1
# zfs set dept:users=general userpool/user2
# zfs set dept:users=itops userpool/user3
```

Toutes les commandes fonctionnant avec des propriétés (par exemple, les commandes `zfs list`, `zfs get`, `zfs set`, etc.) permettent d'utiliser des propriétés natives et des propriétés définies par l'utilisateur.

Par exemple :

```
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
```

Pour supprimer une propriété définie par l'utilisateur, utilisez la commande `zfs inherit`. Par exemple :

```
# zfs inherit -r dept:users userpool
```

Si cette propriété n'est définie dans aucun jeu de données parent, elle est définitivement supprimée.

Envoi de requêtes sur les informations des systèmes de fichiers ZFS

La commande `zfs list` contient un mécanisme extensible permettant d'afficher et d'envoyer des requêtes sur les informations des systèmes de fichiers. Cette section décrit les requêtes de base ainsi que les requêtes plus complexes.

Affichage des informations de base des systèmes ZFS

La commande `zfs list` spécifiée sans option permet de répertorier les informations de base sur les jeux de données. Cette commande affiche le nom de tous les jeux de données définis sur le système ainsi que les valeurs `used`, `available`, `referenced` et `mountpoint` correspondantes. Pour plus d'informations sur ces propriétés, reportez-vous à la section "Présentation des propriétés ZFS" à la page 183.

Par exemple :

```
# 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
```

Cette commande permet d'afficher des jeux de données spécifiques. Pour cela, spécifiez le nom du ou des jeux de données à afficher sur la ligne de commande. Vous pouvez également spécifier l'option `-r` pour afficher de manière récursive tous les descendants des jeux de données. Par exemple :

```
# 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  -
```

Vous pouvez utiliser la commande `zfs list` avec le point de montage d'un système de fichiers. Par exemple :

```
# zfs list /user/home/mark
NAME             USED  AVAIL  REFER  MOUNTPOINT
users/home/mark  1.00G  64.9G  1.00G  /users/home/mark
```

L'exemple suivant montre comment afficher des informations de base sur `tank/home/gina` et l'ensemble de ses systèmes de fichiers descendants :

```
# 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
```

Pour plus d'informations sur la commande `zfs list`, reportez-vous à la page de manuel `zfs(1M)`.

Création de requêtes ZFS complexes

Les options `o`, `-t` et `-H` permettent de personnaliser la sortie de la commande `-zfs list`.

Vous pouvez également personnaliser la sortie des valeurs de propriété en spécifiant l'option `-o` ainsi que la liste des propriétés souhaitées séparées par une virgule. Toute propriété de jeu de données peut être utilisée en tant qu'argument valide. Pour consulter la liste de toutes les propriétés de jeu de données prises en charge, reportez-vous à la section "Présentation des propriétés ZFS" à la page 183. Outre les propriétés répertoriées dans cette section, la liste de l'option `-o` peut également contenir la valeur littérale `name` afin de définir l'inclusion du nom de jeu de données dans la sortie.

L'exemple suivant illustre l'utilisation de la commande `zfs list` pour afficher le nom de jeu de données et des valeurs de propriété `sharenfs` et `mountpoint`.

```
# 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
```

L'option `-t` permet de spécifier le type de jeu de données à afficher. Le tableau suivant décrit les différents types valides.

TABLEAU 5–2 Types d'objets ZFS

Type	Description
`filesystem`	Systèmes de fichiers et clones
`volume`	Volumes
`share`	Partage de système de fichiers
`snapshot`	Instantanés

L'option `-t` permet de spécifier la liste des types de jeux de données à afficher, séparés par une virgule. L'exemple suivant illustre l'affichage du nom et de la propriété `-used` de l'ensemble des systèmes de fichiers via l'utilisation simultanée des options `-t` et `o` :

```
# 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
```

L'option `-H` permet d'exclure l'en-tête de la commande `zfs list` lors de la génération de la sortie. L'option `-H` permet de remplacer les espaces par un caractère de tabulation. Cette option permet notamment d'effectuer des analyses sur les sorties (par exemple, des scripts). L'exemple suivant illustre la sortie de la commande `zfs list` spécifiée avec l'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
```

Gestion des propriétés ZFS

La gestion des propriétés de jeu de données s'effectue à l'aide des sous-commandes `set`, `inherit` et `get` de la commande `zfs`.

- "Définition des propriétés ZFS" à la page 200
- "Héritage des propriétés ZFS" à la page 201
- "Envoi de requêtes sur les propriétés ZFS" à la page 202

Définition des propriétés ZFS

La commande `zfs set` permet de modifier les propriétés de jeu de données pouvant être définies. Vous pouvez également définir les propriétés lors de la création des jeux de données à l'aide de la commande `zfs create`. Pour consulter la listes des propriétés de jeu de données définies, reportez-vous à la section "Propriétés ZFS natives définies" à la page 193.

La commande `zfs set` permet d'indiquer une séquence propriété/valeur au format *property=value*, suivie du nom du jeu de données. Lors de chaque appel de la commande `zfs set`, vous ne pouvez définir ou modifier qu'une propriété à la fois.

L'exemple suivant illustre la définition de la propriété `atime` sur la valeur `off` pour `tank/home`.

```
# zfs set atime=off tank/home
```

Vous pouvez également définir les propriétés des systèmes de fichiers une fois ces derniers créés. Par exemple :

```
# zfs create -o atime=off tank/home
```

Vous pouvez spécifier des valeurs de propriété numériques en utilisant les suffixes faciles à utiliser suivants (par taille croissante) : `BKMGTPEZ`. Ces suffixes peuvent être suivis de la lettre `b` (signifiant "byte", octet) à l'exception du suffixe `B`, qui fait déjà référence à cette unité de mesure. Les quatre invocations suivantes de `zfs set` sont des expressions numériques équivalentes qui définissent la propriété `quota` sur la valeur de 20 Go sur le système de fichiers `users/home/mark` :

```
# 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
```

Si vous tentez de définir une propriété sur un système de fichiers à 100% de sa capacité, un message semblable à celui-ci s'affichera :

```
# zfs set quota=20gb users/home/mark
cannot set property for '/users/home/mark': out of space
```

Les valeurs des propriétés non numériques prennent en charge la distinction majuscules/minuscules et doivent être spécifiées sous la forme de minuscules, sauf pour les propriétés `mountpoint` et `sharenfs`. Les valeurs de ces propriétés peuvent utiliser des minuscules et des majuscules.

Pour plus d'informations sur la commande `zfs set`, reportez-vous à la page de manuel `zfs(1M)`.

Héritage des propriétés ZFS

A l'exception des quotas et des réservations, toutes les propriétés pouvant être définies héritent de la valeur du système de fichiers parent (sauf si un quota ou une réservation est explicitement défini sur le système de fichiers descendant). Si aucune valeur explicite n'est définie pour une propriété d'un système ascendant, la valeur par défaut de cette propriété est appliquée. Vous pouvez utiliser la commande `zfs inherit` pour effacer la valeur d'une propriété et faire ainsi hériter la valeur du système de fichiers parent.

L'exemple suivant utilise la commande `zfs set` pour activer la compression du système de fichiers `tank/home/jeff`. La commande `zfs inherit` est ensuite exécutée afin de supprimer la valeur de la propriété `compression`, entraînant ainsi l'héritage de la valeur par défaut `off`. En effet, la propriété `compression` n'est définie localement ni pour `home`, ni pour `tank` ; la valeur par défaut est donc appliquée. Si la compression avait été activée pour ces deux systèmes, la valeur définie pour le système ascendant direct aurait été utilisée (en l'occurrence, `home`).

```
# 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
```

La sous-commande `inherit` est appliquée de manière récursive lorsque l'option `-r` est spécifiée. Dans l'exemple suivant, la commande entraîne l'héritage de la valeur de la propriété `compression` pour `tank/home` ainsi que pour ses éventuels descendants :

```
# zfs inherit -r compression tank/home
```

Remarque – L'utilisation de l'option `-r` supprime la valeur de propriété actuelle pour l'ensemble des systèmes de fichiers descendants.

Pour plus d'informations sur la commande `zfs inherit`, reportez-vous à la page de manuel `zfs`(1M).

Envoi de requêtes sur les propriétés ZFS

Le moyen le plus simple pour envoyer une requête sur les valeurs de propriété consiste à exécuter la commande `zfs list`. Pour plus d'informations, reportez-vous à la section “Affichage des informations de base des systèmes ZFS” à la page 198. Cependant, dans le cadre de requêtes complexes et pour les scripts, utilisez la commande `zfs get` afin de fournir des informations plus détaillées dans un format personnalisé.

La commande `zfs get` permet de récupérer les propriétés de jeu de données. L'exemple suivant illustre la récupération d'une seule valeur de propriété au sein d'un jeu de données :

```
# zfs get checksum tank/ws
NAME            PROPERTY       VALUE                      SOURCE
tank/ws         checksum       on                         default
```

La quatrième colonne `SOURCE` indique l'origine de la valeur de cette propriété. Le tableau ci-dessous définit les valeurs possibles de la source.

TABLEAU 5–3 Valeurs possibles de la colonne SOURCE (commande `zfs get`)

Valeur de source	Description
`default`	Cette propriété n'a jamais été définie de manière explicite pour ce jeu de données ni pour ses systèmes ascendants. La valeur par défaut est utilisée.
`inherited from` *dataset-name*	La valeur de propriété est héritée du jeu de données parent spécifié par la chaîne *dataset-name*.
`local`	La valeur de propriété a été définie de manière explicite pour ce jeu de données à l'aide de la commande `zfs set`.
`temporary`	Cette valeur de propriété a été définie à l'aide la commande `zfs mount` spécifiée avec l'option `-o` et n'est valide que pour la durée du montage. Pour plus d'informations sur les propriétés de point de montage temporaires, reportez-vous à la section "Utilisation de propriétés de montage temporaires" à la page 209.
None (Aucune)	Cette propriété est en lecture seule. Sa valeur est générée par ZFS.

Le mot-clé `all` permet de récupérer toutes les valeurs de propriétés du jeu de données. Les exemples suivants utilisent le mot-clé `all` :

Remarque – Le fonctionnement des propriétés `casesensitivity`, `nbmand`, `normalization`, `sharesmb`, `utf8only` et `vscan` n'est pas optimal dans la version Solaris 10 car le service Oracle Solaris SMB n'est pas pris en charge dans la version Solaris 10.

```
# 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
```

L'option `-s` spécifiée avec la commande `zfs get` permet de spécifier, en fonction du type de source, les propriétés à afficher. Cette option permet d'indiquer la liste des types de sources souhaités, séparés par une virgule. Seules les propriétés associées au type de source spécifié sont affichées. Les types de source valides sont `local`, `default`, `inherited`, `temporary` et `none`. L'exemple suivant indique l'ensemble des propriétés définies localement sur `tank/ws`.

```
# zfs get -s local all tank/ws
NAME     PROPERTY              VALUE                  SOURCE
tank/ws  compression           on                     local
```

Les options décrites ci-dessus peuvent être associées à l'option `-r` afin d'afficher de manière récursive les propriétés spécifiées sur les systèmes enfant du système de fichiers concerné. Dans l'exemple suivant, toutes les propriétés temporaires de l'ensemble des systèmes de fichiers de `tank/home` sont affichées de façon récursive :

```
# 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
```

Vous pouvez interroger les valeurs d'une propriété à l'aide de la commande `zfs get` sans spécifier le système de fichiers cible (la commande fonctionne sur tous les systèmes de fichiers et les pools). Par exemple :

```
# zfs get -s local all
tank/home               atime          off                   local
tank/home/jeff          atime          off                   local
tank/home/mark          quota          20G                   local
```

Pour plus d'informations sur la commande `zfs get`, reportez-vous à la page de manuel `zfs(1M)`.

Envoi de requête sur les propriétés ZFS pour l'exécution de scripts

La commande `zfs get` prend en charge les options `-H` et `-o`, qui permettent l'exécution de scripts. Vous pouvez utiliser l'option `-H` pour omettre les informations d'en-tête et pour remplacer un espace par un caractère de tabulation. L'uniformisation des espaces permet de faciliter l'analyse des données. Vous pouvez utiliser l'option `-o` pour personnaliser la sortie de l'une des façons suivantes :

- Le littéral `nom` peut être utilisé avec une liste séparée par des virgules de propriétés comme l'explique la section "Présentation des propriétés ZFS" à la page 183.
- Une liste de champs littéraux séparés par des virgules (`name`, `value`, `property` et `source`) suivi d'un espace et d'un argument. En d'autres termes, il s'agit d'une liste de propriétés séparées par des virgules.

L'exemple suivant illustre la commande permettant de récupérer une seule valeur en spécifiant les options `-H` et `-o` de la commande `zfs get`:

```
# zfs get -H -o value compression tank/home
on
```

L'option `-p` indique les valeurs numériques sous leur forme exacte. Par exemple, 1 Mo serait signalé sous la forme 1000000. Cette option peut être utilisée comme suit :

```
# zfs get -H -o value -p used tank/home
182983742
```

L'option `-r` permet de récupérer de manière récursive les valeurs demandées pour l'ensemble des descendants et peut s'utiliser avec toutes les options mentionnées précédemment. Dans l'exemple suivant, les options `-H`, `-o` et `-r` sont spécifiées afin de récupérer le nom du système de fichiers et la valeur de la propriété `used` pour `export/home` et ses descendants, tout en excluant les en-têtes dans la sortie :

```
# zfs get -H -o name,value -r used export/home
```

Montage de système de fichiers ZFS

Cette section décrit le processus de montage des systèmes de fichiers ZFS

- "Gestion des points de montage ZFS" à la page 206
- "Montage de système de fichiers ZFS" à la page 208
- "Utilisation de propriétés de montage temporaires" à la page 209
- "Démontage des systèmes de fichiers ZFS" à la page 209

Gestion des points de montage ZFS

Par défaut, un système de fichiers ZFS est automatiquement monté lors de sa création. Vous pouvez déterminer un comportement de point de montage spécifique pour un système de fichiers comme décrit dans cette section.

Vous pouvez également définir le point de montage par défaut du système de fichiers d'un pool lors de l'exécution de la commande de création `zpool create` en spécifiant l'option `-m`. Pour plus d'informations sur la création de pools, reportez-vous à la section "Création de pools de stockage ZFS" à la page 52.

Tous les systèmes de fichiers ZFS sont montés lors de l'initialisation à l'aide du service `svc://system/filesystem/local` SMF (Service Management Facility). Les systèmes de fichiers sont montés sous */path*, où *path* correspond au nom du système de fichiers. Système de fichiers ZFS

Vous pouvez remplacer le point de montage par défaut à l'aide de la commande `zfs set` pour définir la propriété `mountpoint` sur un chemin spécifique. ZFS crée automatiquement le point de montage spécifié, si nécessaire, et monte automatiquement le système de fichiers correspondant.

Les systèmes de fichiers ZFS sont automatiquement montés au moment de l'initialisation sans qu'il soit nécessaire d'éditer le fichier `/etc/vfstab`.

La propriété `mountpoint` est héritée. Par exemple, si le fichier `pool/home` est doté d'une propriété `mountpoint` définie sur `/export/stuff`, alors `pool/home/user` hérite de la valeur `/export/stuff/user` pour sa propriété `mountpoint`.

Pour éviter le montage d'un système de fichiers, définissez la propriété `mountpoint` sur `none`. En outre, la propriété `canmount` peut être utilisée pour contrôler le montage d'un système de fichiers. Pour plus d'informations sur la propriété `canmount`, reportez-vous à la section "Propriété `canmount`" à la page 195.

Les systèmes de fichiers peuvent également être gérés de manière explicite à l'aide d'interfaces de montage héritées en utilisant la commande `zfs set` pour définir la propriété `mountpoint` sur `legacy`. Dans ce cas, le montage et la gestion d'un système de fichiers ne sont pas gérés automatiquement par ZFS. Ces opérations s'effectuent alors à l'aide des outils hérités, comme les commandes `mount` et `umount` et le fichier `/etc/vfstab`. Pour plus d'informations sur les montages hérités, reportez-vous à la section "Points de montage hérités" à la page 207.

Points de montage automatiques

- Lorsque vous modifiez la propriété `mountpoint` de `legacy` à `none` sur un chemin spécifique, le système de fichiers ZFS est automatiquement monté.
- Si le système de fichiers ZFS est géré automatiquement sans être monté et si la propriété `mountpoint` est modifiée, le système de fichiers reste démonté.

Les systèmes de fichiers dont la propriété `mountpoint` n'est pas définie sur `legacy` sont gérés par le système ZFS. L'exemple suivant illustre la création d'un système de fichiers dont le point de montage est automatiquement géré par le système ZFS :

```
# 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                       -
```

Vous pouvez également définir la propriété `mountpoint` de manière explicite, comme dans l'exemple suivant :

```
# zfs set mountpoint=/mnt pool/filesystem
# zfs get mountpoint pool/filesystem
NAME              PROPERTY      VALUE                     SOURCE
pool/filesystem  mountpoint    /mnt                      local
# zfs get mounted pool/filesystem
NAME              PROPERTY      VALUE                     SOURCE
pool/filesystem  mounted       yes                       -
```

Si la propriété `mountpoint` est modifiée, le système de fichiers est automatiquement démonté de l'ancien point de montage et remonté sur le nouveau. Les répertoires de point de montage sont créés, le cas échéant. Si ZFS n'est pas en mesure de démonter un système de fichiers parce qu'il est actif, une erreur est signalée et un démontage manuel forcé doit être effectué.

Points de montage hérités

La gestion des systèmes de fichiers ZFS peut s'effectuer à l'aide d'outils hérités. Pour cela, la propriété `mountpoint` doit être définie sur `legacy`. Les systèmes de fichiers hérités sont alors gérés à l'aide des commandes `mount` et `umount` et du fichier `/etc/vfstab`. Lors de l'initialisation, le système de fichiers ZFS ne monte pas automatiquement les systèmes de fichiers hérités et les commandes ZFS `mount` et `umount` ne fonctionnent pas sur ces types de systèmes de fichiers. Les exemples suivants illustrent la définition et la gestion d'un système de fichiers ZFS hérité :

```
# zfs set mountpoint=legacy tank/home/eric
# mount -F zfs tank/home/eschrock /mnt
```

Pour monter automatiquement un système de fichiers hérité lors de l'initialisation, vous devez ajouter une entrée au fichier `/etc/vfstab`. L'exemple suivant montre l'entrée telle qu'elle peut apparaître dans le fichier `/etc/vfstab` :

```
#device         device        mount           FS      fsck    mount   mount
#to mount       to fsck       point           type    pass    at boot options
#

tank/home/eric  -        /mnt           zfs        -        yes        -
```

Les entrées `device to fsck` et `fsck pass` sont définies sur `-` car la commande `fsck` ne s'applique pas aux systèmes de fichiers ZFS. Pour plus d'informations sur l'intégrité des données ZFS, reportez-vous à la section "Sémantique transactionnelle" à la page 26.

Montage de système de fichiers ZFS

Le montage des systèmes de fichiers ZFS s'effectue automatiquement lors du processus de création ou lors de l'initialisation du système. Vous ne devez utiliser la commande `zfs mount` que lorsque vous devez modifier les options de montage ou monter/démonter explicitement les systèmes de fichiers.

Spécifiée sans argument, la commande `zfs mount` répertorie tous les systèmes de fichiers actuellement montés gérés par ZFS. Les points de montage hérités ne sont pas inclus. Par exemple :

```
# zfs mount | grep tank/home
zfs mount | grep tank/home
tank/home                       /tank/home
tank/home/jeff                  /tank/home/jeff
```

L'option `-a` permet de monter tous les systèmes de fichiers ZFS. Les systèmes de fichiers hérités ne sont pas montés. Par exemple :

```
# zfs mount -a
```

Par défaut, le système ZFS autorise uniquement le montage sur les répertoires vides. Par exemple :

```
# zfs mount tank/home/lori
cannot mount 'tank/home/lori': filesystem already mounted
```

La gestion des points de montage hérités doit s'effectuer à l'aide des outils hérités. Toute tentative d'utilisation des outils ZFS génère une erreur. Par exemple :

```
# 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
```

Le montage d'un système de fichiers requiert l'utilisation d'un ensemble d'options basées sur les valeurs des propriétés associées au système de fichiers. Le tableau ci-dessous illustre la corrélation entre les propriétés et les options de montage :

TABLEAU 5–4 Options et propriétés de montage ZFS

Propriété	Option de montage
`atime`	`atime/noatime`

TABLEAU 5–4 Options et propriétés de montage ZFS *(Suite)*

Propriété	Option de montage
devices	devices/nodevices
exec	exec/noexec
nbmand	nbmand/nonbmand
readonly	ro/rw
setuid	setuid/nosetuid
xattr	xattr/noaxttr

L'option de montage `nosuid` représente un alias de `nodevices,nosetuid`.

Utilisation de propriétés de montage temporaires

Si les options de montage décrites à la section précédente sont définies de manière explicite en spécifiant l'option `-o` avec la commande `zfs mount`, les valeurs des propriétés associées sont remplacées de manière temporaire. Ces valeurs de propriété sont désignées par la chaîne `temporary` dans la commande `zfs get` et reprennent leur valeur d'origine une fois le système de fichiers démonté. Si une valeur de propriété est modifiée alors que le système de fichiers est monté, la modification prend immédiatement effet et remplace toute valeur temporaire.

L'exemple suivant illustre la définition temporaire de l'option de montage en lecture seule sur le système de fichiers `tank/home/neil`. Le système de fichiers est censé être démonté.

```
# zfs mount -o ro users/home/neil
```

Pour modifier temporairement une valeur de propriété sur un système de fichiers monté, vous devez utiliser l'option spécifique `remount`. Dans l'exemple suivant, la propriété `atime` est temporairement définie sur la valeur `off` pour un système de fichiers monté :

```
# zfs mount -o remount,noatime users/home/neil
NAME             PROPERTY  VALUE  SOURCE
users/home/neil  atime     off    temporary
# zfs get atime users/home/perrin
```

Pour plus d'informations sur la commande `zfs mount`, reportez-vous à la page de manuel `zfs(1M)`.

Démontage des systèmes de fichiers ZFS

Le démontage des systèmes de fichiers ZFS peut s'effectuer à l'aide de la commande `zfs unmount`. La commande `unmount` peut utiliser le point de montage ou le nom du système de fichiers comme argument.

L'exemple suivant illustre le démontage d'un système de fichiers avec l'argument de nom de système de fichiers :

```
# zfs unmount users/home/mark
```

L'exemple suivant illustre le démontage d'un système de fichiers avec l'argument de point de montage :

```
# zfs unmount /users/home/mark
```

Si le système de fichiers est occupé, la commande `unmount` échoue. L'option `-f` permet de forcer le démontage d'un système de fichiers. Le démontage forcé d'un système de fichiers requiert une attention particulière si le contenu de ce système est en cours d'utilisation. Ce type d'opération peut entraîner des comportements d'application imprévisibles.

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

Pour garantir la compatibilité ascendante, vous pouvez démonter les systèmes de fichiers ZFS à l'aide de la commande héritée `umount`. Par exemple :

```
# umount /tank/home/bob
```

Pour plus d'informations sur la commande `zfs unmount`, reportez-vous à la page de manuel `zfs(1M)`.

Activation et annulation du partage des systèmes de fichiers ZFS

Le système de fichiers ZFS permet de partager automatiquement des systèmes de fichiers en définissant la propriété `sharenfs`. Cette propriété évite d'avoir à modifier le fichier `/etc/dfs/dfstab` lors de l'ajout d'un système de fichiers. La propriété `sharenfs` correspond à une liste d'options séparées par une virgule spécifiée avec la commande `share.` La valeur `on` constitue un alias des options de partage par défaut, qui attribuent les autorisations `read/write` à tous les utilisateurs. La valeur `off` indique que le partage du système de fichiers n'est pas géré par ZFS et qu'il s'effectue à l'aide des outils classiques (par exemple, à l'aide du fichier `/etc/dfs/dfstab`). Tous les systèmes de fichiers dont la propriété `sharenfs` n'est pas définie sur `off` sont partagés lors de l'initialisation.

Contrôle de la sémantique de partage

Par défaut, aucun système de fichiers n'est partagé. Pour partager un nouveau système de fichiers, utilisez une syntaxe de la commande `zfs set` similaire à celle présentée à l'exemple ci-dessous :

```
# zfs set sharenfs=on tank/home/eric
```

La propriété `sharenfs` est héritée et les systèmes de fichiers sont automatiquement partagés lorsqu'ils sont créés (si la propriété héritée n'est pas définie sur `off`). Exemple :

```
# zfs set sharenfs=on tank/home
# zfs create tank/home/bill
# zfs create tank/home/mark
# zfs set sharenfs=ro tank/home/bob
```

`tank/home/bill` et `tank/home/mark` sont partagés en écriture dès leur création, car ils héritent de la propriété `sharenfs` de `tank/home`. Une fois la propriété définie sur `ro` (lecture seule), `tank/home/mark` est partagé en lecture seule, quelle que soit la propriété `sharenfs` définie pour `tank/home` .

Annulation du partage des systèmes de fichiers ZFS

Même si la plupart des systèmes de fichiers sont automatiquement partagés ou non lors de l'initialisation, la création ou la destruction de ces derniers, il est parfois nécessaire de les définir explicitement comme non partagés. Ce type d'opération s'effectue à l'aide de la commande `zfs unshare.` Par exemple :

```
# zfs unshare tank/home/mark
```

Cette commande entraîne l'annulation du partage du système de fichiers `tank/home/mark`. Pour annuler le partage de tous les systèmes de fichiers ZFS du système, vous devez utiliser l'option `-a`.

```
# zfs unshare -a
```

Partage des systèmes de fichiers ZFS

La plupart du temps, le comportement automatique du système de fichiers ZFS par rapport au partage de système de fichiers lors de l'initialisation et de la création de ces derniers est suffisant pour les opérations normales. Si vous êtes amené à annuler le partage d'un système de fichiers, vous pouvez réactiver le partage à l'aide de la commande `zfs share.` Par exemple :

```
# zfs share tank/home/mark
```

L'option `-a` permet également de partager tous les systèmes de fichiers ZFS du système.

```
# zfs share -a
```

Comportement de partage hérité

Si la propriété `sharenfs` est définie sur `off`, l'activation et l'annulation du partage des systèmes de fichiers ZFS sont désactivées de manière permanente. Cette valeur permet de gérer le partage des systèmes de fichiers à l'aide des outils classiques (par exemple, à l'aide du fichier `/etc/dfs/dfstab`).

Contrairement à la commande `mount` héritée, les commandes `share` et `unshare` peuvent être exécutées sur les systèmes de fichiers ZFS. Vous pouvez dès lors partager manuellement un système de fichiers en spécifiant des options différentes de celles définies pour la propriété `sharenfs`. Ce modèle de gestion est déconseillé. La gestion des partages NFS doit s'effectuer intégralement à l'aide du système ZFS ou intégralement à l'aide du fichier `/etc/dfs/dfstab`. Les modèle ZFS a été conçu pour simplifier et pour faciliter les opérations de gestion par rapport au modèle classique.

Définition des quotas et réservations ZFS

La propriété `quota` permet de limiter la quantité d'espace disque disponible pour un système de fichiers. La propriété `reservation` permet quant à elle de garantir la disponibilité d'une certaine quantité d'espace disque pour un système de fichiers. Ces deux propriétés s'appliquent au système de fichiers sur lequel elles sont définies ainsi qu'à ses descendants.

Par exemple, si un quota est défini pour le système de fichiers `tank/home`, la quantité totale d'espace disque utilisée par `tank/home` *et l'ensemble de ses descendants* ne peut pas excéder le quota défini. De même, si une réservation est définie pour le jeu de données `tank/home`, cette réservation s'applique à `tank/home` *et à tous ses descendants*. La quantité d'espace disque utilisée par un système de fichiers et par tous ses descendants est indiquée par la propriété `used`.

Les propriétés `refquota` et `refreservation` vous permettent de gérer l'espace d'un système de fichiers sans prendre en compte l'espace disque utilisé par les descendants, notamment les instantanés et les clones.

Dans cette version de Solaris, vous pouvez définir un quota d'utilisateur (*user*) ou de *groupe* sur la quantité d'espace disque utilisée par les fichiers appartenant à un utilisateur ou à un groupe spécifique. Les propriétés de quota d'utilisateur et de groupe ne peuvent pas être définies sur un volume, sur un système de fichiers antérieur à la version 4, ou sur un pool antérieur à la version 15.

Considérez les points suivants pour déterminer quelles fonctions de quota et de réservation conviennent le mieux à la gestion de vos systèmes de fichiers :

- Les propriétés `quota` et `reservation` conviennent à la gestion de l'espace disque utilisé par les systèmes de fichiers et leurs descendants.
- Les propriétés `refquota` et `refreservation` conviennent à la gestion de l'espace disque utilisé par les systèmes de fichiers.
- La définition d'une propriété `refquota` ou `refreservation` supérieure à une la propriété `quota` ou `reservation` n'a aucun effet. Lorsque vous définissez la propriété `quota` ou `refquota`, les opérations qui tentent de dépasser l'une de ces valeurs échouent. Il est possible de dépasser une valeur `quota` supérieure à une valeur `refquota`. Par exemple, si certains blocs d'instantanés sont modifiés, la valeur `quota` risque d'être dépassée avant la valeur `refquota`.

- Les quotas d'utilisateurs et de groupes permettent d'augmenter plus facilement l'espace disque contenant de nombreux comptes d'utilisateur, par exemple dans une université.

Pour plus d'informations sur la définition de quotas et réservations, reportez-vous aux sections "Définitions de quotas sur les systèmes de fichiers ZFS" à la page 213 et "Définition de réservations sur les systèmes de fichiers ZFS" à la page 216.

Définitions de quotas sur les systèmes de fichiers ZFS

Les quotas des systèmes de fichiers ZFS peuvent être définis et affichés à l'aide des commandes `zfs set` et `zfs get`. Dans l'exemple suivant, un quota de 10 Go est défini pour `tank/home/jeff` :

```
# zfs set quota=10G tank/home/jeff
# zfs get quota tank/home/jeff
NAME            PROPERTY  VALUE  SOURCE
tank/home/jeff  quota     10G    local
```

Les quotas affectent également la sortie des commandes `zfs list` et `df`. Par exemple :

```
# 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
```

`tank/home` dispose de 66,9 Go d'espace disque disponible. Toutefois, `tank/home/jeff` et `tank/home/jeff/ws` disposent uniquement de 10 Go d'espace disponible, respectivement, en raison du quota défini pour `tank/home/jeff`.

Vous ne pouvez pas définir un quota sur une valeur inférieure à la quantité d'espace actuellement utilisée par un système de fichiers. Par exemple :

```
# zfs set quota=10K tank/home/jeff
cannot set property for 'tank/home/jeff':
size is less than current used or reserved space
```

Vous pouvez définir une propriété `refquota` sur un système de fichiers pour limiter l'espace disque occupé par le système de fichiers. Cette limite fixe ne comprend pas l'espace disque utilisé par les descendants. Par exemple, le quota de 10 Go de studentA n'est pas affecté par l'espace utilisé par les instantanés.

```
# 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
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  -
```

Par souci de commodité, vous pouvez définir un autre quota pour un système de fichiers afin de vous aider à gérer l'espace disque utilisé par les instantanés. Par exemple :

```
# 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  -
```

Dans ce scénario, studentA peut atteindre la limite maximale de refquota (10 Go), mais studentA peut supprimer des fichiers pour libérer de l'espace même en présence d'instantanés.

Dans l'exemple précédent, le plus petit des deux quotas (10 Go par rapport à 20 Go) s'affiche dans la sortie zfs list. Pour afficher la valeur des deux quotas, utilisez la commande zfs get. Par exemple :

```
# zfs get refquota,quota students/studentA
NAME               PROPERTY  VALUE              SOURCE
students/studentA  refquota  10G                local
students/studentA  quota     20G                local
```

Définition de quotas d'utilisateurs et de groupes sur un système de fichiers ZFS

Vous pouvez définir un quota d'utilisateurs ou de groupes en utilisant respectivement les commandes zfs userquota et zfs groupquota. Par exemple :

```
# zfs create students/compsci
# zfs set userquota@student1=10G students/compsci
# zfs create students/labstaff
# zfs set groupquota@labstaff=20GB students/labstaff
```

Affichez le quota d'utilisateurs ou de groupes actuel comme suit :

```
# 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
```

Vous pouvez afficher l'utilisation générale de l'espace disque par les utilisateurs et les groupes en interrogeant les propriétés suivantes :

```
# 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
```

Pour identifier l'utilisation de l'espace disque d'un groupe ou d'un utilisateur, vous devez interroger les propriétés suivantes :

```
# 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
```

Les propriétés de quota d'utilisateurs et de groupes ne sont pas affichées à l'aide de la commande `zfs get all` *dataset*, qui affiche une liste de toutes les autres propriétés du système de fichiers.

Vous pouvez supprimer un quota d'utilisateurs ou de groupes comme suit :

```
# zfs set userquota@student1=none students/compsci
# zfs set groupquota@labstaff=none students/labstaff
```

Les quotas d'utilisateurs et de groupes sur les systèmes de fichiers ZFS offrent les fonctionnalités suivantes :

- Un quota d'utilisateurs ou de groupes défini sur un système de fichiers parent n'est pas automatiquement hérité par un système de fichiers descendant.
- Cependant, le quota d'utilisateurs ou de groupes est appliqué lorsqu'un clone ou un instantané est créé à partir d'un système de fichiers lié à un quota d'utilisateurs ou de groupes. De même, un quota d'utilisateurs ou de groupes est inclus avec le système de fichiers lorsqu'un flux est créé à l'aide de la commande `zfs send`, même sans l'option `-R`.
- Les utilisateurs dénués de privilèges peuvent uniquement disposer de leur propre utilisation d'espace disque. L'utilisateur root ou l'utilisateur qui s'est vu accorder le privilège `userused` ou `groupused` peut accéder aux informations de comptabilité de l'espace disque utilisateur ou groupe de tout le monde.
- Les propriétés `userquota` et `groupquota` ne peuvent pas être définies sur les volumes ZFS, sur un système de fichiers antérieur à la version 4, ou sur un pool antérieur à la version 15.

L'application des quotas d'utilisateurs et de groupes peut être différée de quelques secondes. Ce délai signifie que les utilisateurs peuvent dépasser leurs quotas avant que le système ne le remarque et refuse d'autres écritures en affichant le message d'erreur `EDQUOT` .

Vous pouvez utiliser la commande quota héritée pour examiner les quotas d'utilisateurs dans un environnement NFS où un système de fichiers ZFS est monté, par exemple. Sans aucune option, la commande quota affiche uniquement la sortie en cas de dépassement du quota de l'utilisateur. Par exemple :

```
# 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
```

Si vous réinitialisez le quota d'utilisateurs et que la limite du quota n'est plus dépassée, vous devez utiliser la commande quota -v pour examiner le quota de l'utilisateur. Par exemple :

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

Définition de réservations sur les systèmes de fichiers ZFS

Une *réservation* ZFS désigne une quantité d'espace disque du pool garantie pour un jeu de données. Dès lors, pour réserver une quantité d'espace disque pour un jeu de données, cette quantité doit être actuellement disponible sur le pool. La quantité totale d'espace non utilisé des réservations ne peut pas dépasser la quantité d'espace disque non utilisé du pool. La définition et l'affichage des réservations ZFS s'effectuent respectivement à l'aide des commandes zfs set et zfs get. Par exemple :

```
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
NAME            PROPERTY     VALUE   SOURCE
tank/home/bill  reservation  5G      local
```

Les réservations peuvent affecter la sortie de la commande zfs list. Par exemple :

```
# 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
```

Notez que `tank/home` utilise 5 Go d'espace bien que la quantité totale d'espace à laquelle `tank/home` et ses descendants font référence est bien inférieure à 5 Go. L'espace utilisé reflète l'espace réservé pour `tank/home/bill`. Les réservations sont prises en compte dans le calcul de l'espace disque utilisé du système de fichiers parent et non dans le quota, la réservation ou les deux.

```
# zfs set quota=5G pool/filesystem
# zfs set reservation=10G pool/filesystem/user1
cannot set reservation for 'pool/filesystem/user1': size is greater than
available space
```

Un jeu de données peut utiliser davantage d'espace disque que sa réservation, du moment que le pool dispose d'un espace non réservé et disponible et que l'utilisation actuelle du jeu de données se trouve en dessous des quotas. Un jeu de données ne peut pas utiliser un espace disque réservé à un autre jeu de données.

Les réservations ne sont pas cumulatives. En d'autres termes, l'exécution d'une nouvelle commande `zfs set` pour un jeu de données déjà associé à une réservation n'entraîne pas l'ajout de la nouvelle réservation à la réservation existante. La seconde réservation remplace la première. Par exemple :

```
# 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
```

Vous pouvez définir une réservation `refreservation` pour garantir un espace disque ne contenant aucun instantané ou clone au jeu de données. Cette valeur est prise en compte dans le calcul de l'espace utilisé des jeux de données parent et vient en déduction des quotas et réservations des jeux de données parent. Par exemple :

```
# 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
```

Vous pouvez également définir une valeur de réservation pour le même jeu de données afin de garantir l'espace du jeu de données et pas de l'espace des instantanés. Par exemple :

```
# 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
```

Les réservations régulières sont prises en compte dans le calcul de l'espace utilisé du parent.

Dans l'exemple précédent, le plus petit des deux quotas (10 Go par rapport à 20 Go) s'affiche dans la sortie `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. Par exemple :

```
# zfs get reservation,refreserv profs/prof1
NAME         PROPERTY        VALUE        SOURCE
profs/prof1  reservation     20G          local
profs/prof1  refreservation  10G          local
```

Lorsque la propriété `refreservation` est définie, un instantané n'est autorisé que si suffisamment d'espace non réservé est disponible dans le pool au-delà de cette réservation afin de pouvoir contenir le nombre actuel d'octets *référencés* dans le jeu de données.

Mise à niveau des systèmes de fichiers ZFS

Si vous possédez des systèmes de fichiers ZFS d'une version antérieure de Solaris, vous pouvez procéder à la mise à niveau de vos systèmes de fichiers à l'aide de la commande `zfs upgrade` afin de tirer parti des fonctions du système de fichiers dans la version actuelle. De plus, cette commande vous avertit lorsque vos systèmes de fichiers exécutent des versions antérieures.

Par exemple, la version de ce système de fichiers est la version 5 actuelle.

```
# zfs upgrade
This system is currently running ZFS filesystem version 5.

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

Utilisez cette commande pour identifier les fonctions disponibles pour chaque version des systèmes de fichiers.

```
# 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.
```

◆ ◆ ◆ CHAPITRE 6

Utilisation des instantanés et des clones ZFS Oracle Solaris

Ce chapitre fournit des informations sur la création et la gestion d'instantanés et de clones ZFS Oracle Solaris. Des informations concernant l'enregistrement des instantanés sont également fournies.

Ce chapitre contient les sections suivantes :

- "Présentation des instantanés ZFS" à la page 219
- "Création et destruction d'instantanés ZFS" à la page 220
- "Affichage et accès des instantanés ZFS" à la page 223
- "Restauration d'un instantané ZFS" à la page 225
- "Présentation des clones ZFS" à la page 227
- "Création d'un clone ZFS" à la page 227
- "Destruction d'un clone ZFS" à la page 228
- "Remplacement d'un système de fichiers ZFS par un clone ZFS" à la page 228
- "Envoi et réception de données ZFS" à la page 229

Présentation des instantanés ZFS

Un *instantané* est une copie en lecture seule d'un système de fichiers ou d'un volume. La création des instantanés est quasiment immédiate. Initialement, elle ne consomme pas d'espace disque supplémentaire au sein du pool. Toutefois, à mesure que les données contenues dans le jeu de données actif changent, l'instantané consomme de l'espace disque en continuant à faire référence aux anciennes données et empêche donc la libération de l'espace disque.

Les instantanés ZFS présentent les caractéristiques suivantes :

- Persistance au cours des réinitialisations de système.
- Théoriquement, le nombre maximal d'instantanés est de 2^{64} instantanés.
- Les instantanés n'utilisent aucune sauvegarde de secours distincte. Les instantanés consomment de l'espace disque provenant directement du pool de stockage auquel appartient le système de fichiers ou le volume à partir duquel ils ont été créés.

- Une seule opération, dite atomique, permet de créer rapidement des instantanés récursifs. Ceux-ci sont tous créés simultanément ou ne sont pas créés du tout. Grâce à ce type d'opération d'instantané atomique, une certaine cohérence des données d'instantané est assurée, y compris pour les systèmes de fichiers descendants.

Il n'est pas possible d'accéder directement aux instantanés de volumes, mais ils peuvent être clonés, sauvegardés, restaurés, etc. Pour plus d'informations sur la sauvegarde d'un instantané ZFS, reportez-vous à la section "Envoi et réception de données ZFS" à la page 229.

- "Création et destruction d'instantanés ZFS" à la page 220
- "Affichage et accès des instantanés ZFS" à la page 223
- "Restauration d'un instantané ZFS" à la page 225

Création et destruction d'instantanés ZFS

La commande `zfs snapshot` permet de créer les instantanés. Elle ne prend pour argument que le nom de l'instantané à créer. Le nom de l'instantané est spécifié comme suit :

filesystem@snapname
volume@snapname

Ce nom doit respecter les conventions d'attribution de nom définies à la section "Exigences d'attribution de noms de composants ZFS" à la page 31.

Dans l'exemple suivant, un instantané de `tank/home/cindy` nommé `friday` est créé.

```
# zfs snapshot tank/home/cindy@friday
```

Vous pouvez créer des instantanés pour tous les systèmes de fichiers descendants à l'aide de l'option `-r`. Par exemple :

```
# 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  -
```

Les propriétés des instantanés ne sont pas modifiables. Les propriétés des jeux de données ne peuvent pas être appliquées à un instantané. Par exemple :

```
# zfs set compression=on tank/home/cindy@friday
cannot set property for 'tank/home/cindy@friday':
this property can not be modified for snapshots
```

La commande `zfs destroy` permet de détruire les instantanés. Par exemple :

```
# zfs destroy tank/home/cindy@friday
```

La destruction d'un jeu de données est impossible s'il existe des instantanés du jeu de données. Par exemple :

```
# 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
```

En outre, si des clones ont été créés à partir d'un instantané, ils doivent être détruits avant que l'instantané ne puisse être détruit.

Pour de plus amples informations sur la sous-commande `destroy`, reportez-vous à la section "Destruction d'un système de fichiers ZFS" à la page 181.

Conservation des clichés ZFS

Si vous disposez de plusieurs stratégies automatiques pour les instantanés pour que l'instantané le plus ancien soit par exemple détruit par la commande `zfs receive` car il n'existe plus du côté de l'envoi, vous pouvez utiliser la fonction de conservation des instantanés.

La *conservation* d'un instantané empêche sa destruction. En outre, cette fonction permet de supprimer un instantané contenant des clones en attendant la suppression du dernier clone à l'aide de la commande `zfs destroy -d`. Chaque instantané est associé à un décompte de référence utilisateur initialisé sur 0 (zéro). Ce nombre augmente de 1 à chaque fois qu'un instantané est conservé et diminue de 1 à chaque fois qu'un instantané conservé est libéré.

Dans la version précédente d'Oracle Solaris, les instantanés pouvaient uniquement être détruits à l'aide de la commande `zfs destroy` s'ils ne contenaient aucun clone. Dans cette version d'Oracle Solaris, les instantanés doivent également renvoyer un décompte de référence utilisateur égal à 0 (zéro).

Vous pouvez conserver un instantané ou un jeu d'instantanés. Par exemple, la syntaxe suivante insère une balise de conservation `keep` sur `citerne/home/cindys/snap@1` :

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

Vous pouvez utiliser l'option `-r` pour conserver récursivement les instantanés de tous les systèmes de fichiers descendants. Par exemple :

```
# zfs snapshot -r tank/home@now
# zfs hold -r keep tank/home@now
```

Cette syntaxe permet d'ajouter une référence `keep` unique à cet instantané ou à ce jeu d'instantanés. Chaque instantané possède son propre espace de noms de balise dans lequel chaque balise de conservation doit être unique. Si un instantané est conservé, les tentatives de destruction de ce dernier à l'aide de la commande `zfs destroy` échoueront. Par exemple :

```
# zfs destroy tank/home/cindy@snap1
cannot destroy 'tank/home/cindy@snap1': dataset is busy
```

Pour détruire un instantané conservé, utilisez l'option `-d`. Par exemple :

```
# zfs destroy -d tank/home/cindy@snap1
```

Utilisez la commande `zfs holds` pour afficher la liste des instantanés conservés. Par exemple :

```
# 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
```

Vous pouvez utiliser la commande `zfs release` pour libérer un instantané ou un jeu d'instantanés conservé. Par exemple :

```
# zfs release -r keep tank/home@now
```

Si l'instantané est libéré, l'instantané peut être détruit à l'aide de la commande `zfs destroy`. Par exemple :

```
# zfs destroy -r tank/home@now
```

Deux nouvelles propriétés permettent d'identifier les informations de conservation d'un instantané :

- La propriété `defer_destroy` est définie sur `on` si l'instantané a été marqué en vue d'une destruction différée à l'aide de la commande `zfs destroy -d`. Dans le cas contraire, la propriété est définie sur `off`.
- La propriété `userrefs` également appelée décompte de *référence utilisateur*, est définie sur le nombre de conservations pour cet instantané.

Renommage d'instantanés ZFS

Vous pouvez renommer les instantanés. Cependant, ils doivent rester dans le même pool et dans le même jeu de données dans lequel il ont été créés. Par exemple :

```
# zfs rename tank/home/cindy@snap1 tank/home/cindy@today
```

En outre, la syntaxe de raccourci suivante est équivalente à la syntaxe précédente :

```
# zfs rename tank/home/cindy@snap1 today
```

L'opération de renommage (`rename`) d'instantané n'est pas prise en charge, car le nom du pool cible et celui du système de fichiers ne correspondent pas au pool et au système de fichiers dans lesquels l'instantané a été créé :

```
# zfs rename tank/home/cindy@today pool/home/cindy@saturday
cannot rename to 'pool/home/cindy@today': snapshots must be part of same
dataset
```

Vous pouvez renommer de manière récursive les instantanés à l'aide de la commande `zfs rename -r`. Par exemple :

```
# 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  -
```

Affichage et accès des instantanés ZFS

Vous pouvez activer ou désactiver l'affichage des listes d'instantanés de la sortie `zfs list` en utilisant la propriété de pool `listsnapshots`. Cette propriété est activée par défaut.

Si vous désactivez cette propriété, vous pouvez utiliser la commande `zfs list -t snapshot` pour afficher les informations relatives à un instantané. Ou activez la propriété de pool `listsnapshots`. Par exemple :

```
# 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
```

Les instantanés des systèmes de fichiers sont accessibles dans le répertoire `.zfs/snapshot` au sein du root du système de fichiers. Par exemple, si `tank/home/cindy` est monté sur `/home/cindy`, les données de l'instantané de `tank/home/cindy@thursday` sont accessibles dans le répertoire `/home/cindy/.zfs/snapshot/thursday`.

```
# ls /tank/home/cindy/.zfs/snapshot
thursday   tuesday    wednesday
```

Vous pouvez répertorier les instantanés comme suit :

```
# 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  -
```

Vous pouvez répertorier les instantanés qui ont été créés pour un système de fichiers particulier comme suit :

```
# 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
```

Comptabilisation de l'espace disque des instantanés ZFS

Lors de la création d'un instantané, son espace disque est initialement partagé entre l'instantané et le système de fichiers et éventuellement avec des instantanés précédents. Lorsque le système de fichiers change, l'espace disque précédemment partagé devient dédié à l'instantané, et il est compté dans la propriété `used` de l'instantané. De plus, la suppression d'instantanés peut augmenter la quantité d'espace disque dédié à d'autres instantanés (et, par conséquent, *utilisé* par ceux-ci).

La valeur de la propriété `referenced` de l'espace d'un instantané est la même que lors de la création de l'instantané sur le système de fichiers.

Vous pouvez identifier des informations supplémentaires sur la façon dont les valeurs de la propriété `used` sont utilisées. Les nouvelles propriétés de système de fichiers en lecture seule décrivent l'utilisation de l'espace disque pour les clones, les systèmes de fichiers et les volumes. Par exemple :

```
$ 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         -       -              -          -
```

Pour une description de ces propriétés, reportez-vous au Tableau 5–1.

Restauration d'un instantané ZFS

Vous pouvez utiliser la commande `zfs rollback` pour abandonner toutes les modifications apportées à un système de fichiers depuis la création d'un instantané spécifique. Le système de fichiers revient à l'état dans lequel il était lors de la prise de l'instantané. Par défaut, la commande ne permet pas de restaurer un instantané autre que le plus récent.

Pour restaurer un instantané précédent, tous les instantanés intermédiaires doivent être détruits. Vous pouvez détruire les instantanés précédents en spécifiant l'option `-r`.

S'il existe des clones d'un instantané intermédiaire, vous devez spécifier l'option `-R` pour détruire également les clones.

Remarque – Si le système de fichiers que vous souhaitez restaurer est actuellement monté, il doit être démonté, puis remonté. Si le système de fichiers ne peut pas être démonté, la restauration échoue. L'option `-f` force le démontage du système de fichiers, le cas échéant.

Dans l'exemple suivant, l'état du système de fichiers `tank/home/cindy` correspondant à l'instantané `tuesday` est restauré.

```
# 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
```

Dans cet exemple, les instantanés `wednesday` et `thursday` sont détruits en raison de la restauration de l'instantané `tuesday` précédent.

```
# zfs list -r -t snapshot -o name,creation tank/home/cindy
NAME                     CREATION
tank/home/cindy@tuesday  Fri Aug  3 15:18 2012
```

Identification des différences entre des instantanés ZFS (`zfs diff`)

Vous pouvez déterminer les différences entre des instantanés ZFS en utilisant la commande `zfs diff`.

Supposons par exemple que les deux instantanés suivants sont créés :

```
$ ls /tank/home/tim
fileA
$ zfs snapshot tank/home/tim@snap1
$ ls /tank/home/tim
fileA  fileB
$ zfs snapshot tank/home/tim@snap2
```

Par exemple, afin d'identifier les différences entre deux instantanés, utilisez une syntaxe semblable à la suivante :

```
$ zfs diff tank/home/tim@snap1 tank/home/tim@snap2
M       /tank/home/tim/
+       /tank/home/tim/fileB
```

Dans la sortie, `M` indique que le répertoire a été modifié. Le `+` indique que `fileB` existe dans l'instantané le plus récent.

Dans la sortie suivante, le `R` indique qu'un fichier dans un instantané a été renommé.

```
$ 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
```

Le tableau suivant résume les modifications apportées au fichier ou au répertoire identifiées par la commande `zfs diff`.

Modification de répertoire ou de fichier	Identificateur
Le fichier ou le répertoire a été modifié ou le lien d'un répertoire ou d'un fichier a changé	`M`
Le fichier ou le répertoire est présent dans l'ancien instantané mais pas dans le plus récent	–
Le fichier ou le répertoire est présent dans l'instantané le plus récent mais pas dans le plus ancien.	`+`
Le fichier ou le répertoire a été renommé	`R`

Pour de plus amples informations, reportez-vous à la page de manuel `zfs(1M)`.

Présentation des clones ZFS

Un *clone* est un volume ou un système de fichiers accessible en écriture et dont le contenu initial est similaire à celui du jeu de données à partir duquel il a été créé. Tout comme pour les instantanés, la création d'un clone est quasiment instantanée et ne consomme initialement aucun espace disque supplémentaire. En outre, vous pouvez prendre un instantané d'un clone.

Les clones se créent uniquement à partir d'un instantané. Lors du clonage d'un instantané, une dépendance implicite se crée entre le clone et l'instantané. Bien que le clone soit créé à une autre emplacement dans la hiérarchie de système de fichiers, l'instantané d'origine ne peut pas être supprimé tant que le clone existe. La propriété `origin` indique cette dépendance et la commande `zfs destroy` répertorie ces dépendances, le cas échéant.

Un clone n'hérite pas des propriétés du jeu de données à partir duquel il a été créé. Les commandes `zfs get` et `zfs set` permettent d'afficher et de modifier les propriétés d'un jeu de données cloné. Pour de plus amples informations sur la configuration des propriétés de jeux de données ZFS, reportez-vous à la section “Définition des propriétés ZFS” à la page 200.

Dans la mesure où un clone partage initialement son espace disque avec l'instantané d'origine, la valeur de la propriété `used` est initialement égale à zéro. A mesure que le clone est modifié, il utilise de plus en plus d'espace disque. La propriété `used` de l'instantané d'origine ne tient pas compte de l'espace disque consommé par le clone.

- “Création d'un clone ZFS” à la page 227
- “Destruction d'un clone ZFS” à la page 228
- “Remplacement d'un système de fichiers ZFS par un clone ZFS” à la page 228

Création d'un clone ZFS

Pour créer un clone, utilisez la commande `zfs clone` en spécifiant l'instantané à partir duquel créer le clone, ainsi que le nom du nouveau volume ou système de fichiers. Le nouveau volume ou système de fichiers peut se trouver à tout emplacement de la hiérarchie ZFS. Le nouveau jeu de données est du même type (un système de fichiers ou un volume, par exemple) que celui de l'instantané à partir duquel le clone a été créé. Vous ne pouvez pas créer le clone d'un système de fichiers dans un autre pool que celui de l'instantané du système de fichiers d'origine.

Dans l'exemple suivant, un nouveau clone appelé `tank/home/matt/bug123` possédant le même contenu initial que l'instantané `tank/ws/gate@yesterday` est créé :

```
# zfs snapshot tank/ws/gate@yesterday
# zfs clone tank/ws/gate@yesterday tank/home/matt/bug123
```

Dans l'exemple suivant, un espace de travail est créé à partir de l'instantané `projects/newproject@today` pour un utilisateur temporaire, sous le nom `projects/teamA/tempuser`. Ensuite, les propriétés sont configurées dans l'espace de travail cloné.

```
# 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
```

Destruction d'un clone ZFS

La commande `zfs destroy` permet de détruire les clones ZFS. Par exemple : Destruction

```
# zfs destroy tank/home/matt/bug123
```

Les clones doivent être détruits préalablement à la destruction de l'instantané parent.

Remplacement d'un système de fichiers ZFS par un clone ZFS

La commande `zfs promote` permet de remplacer un système de fichiers ZFS actif par un clone de ce système de fichiers. Cette fonction facilite le clonage et le remplacement des systèmes de fichiers pour que le système de fichiers *original* devienne le clone du système de fichiers spécifié. En outre, cette fonction permet de détruire le système de fichiers à partir duquel le clone a été créé. Il est impossible de détruire un système de fichiers d'origine possédant des clones actifs, sans le remplacer par l'un de ses clones. Pour plus d'informations sur la destruction des clones, reportez-vous à la section "Destruction d'un clone ZFS" à la page 228.

Dans l'exemple suivant, le système de fichiers `tank/test/productA` est cloné, puis le clone du système de fichiers (`tank/test/productAbeta`) devient le système de fichiers `tank/test/productA` d'origine.

```
# 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  -
```

Dans la sortie `zfs list`, les informations de comptabilisation de l'espace disque du système de fichiers d'origine `productA` ont été remplacées par celles du système de fichiers `productAbeta`

Pour terminer le processus de remplacement de clone, renommez les systèmes de fichiers. Par exemple :

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
```

Vous pouvez également supprimer l'ancien système de fichiers si vous le souhaitez. Par exemple :

```
# zfs destroy tank/test/productAlegacy
```

Envoi et réception de données ZFS

La commande `zfs send` crée une représentation de flux d'un instantané qui est écrite dans la sortie standard. Un flux complet est généré par défaut. Vous pouvez rediriger la sortie vers un fichier ou un système fichier. La commande `zfs receive` crée un instantané dont le contenu est spécifié dans le flux fourni dans l'entrée standard. En cas de réception d'un flux complet, un système de fichiers est également créé. Ces commandes permettent d'envoyer les données d'instantané ZFS et de recevoir les systèmes de fichiers et les données d'instantané ZFS. Reportez-vous aux exemples de la section suivante.

- "Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde" à la page 230
- "Envoi d'un instantané ZFS" à la page 232
- "Réception d'un instantané ZFS" à la page 233
- "Application de différentes valeurs de propriété à un flux d'instantané ZFS" à la page 235
- "Envoi et réception de flux d'instantanés ZFS complexes" à la page 237
- "Réplication distante de données ZFS" à la page 239

Les solutions de sauvegarde suivantes sont disponibles pour enregistrer les données ZFS :

- **Produits de sauvegarde d'entreprise** : si vous souhaitez disposer des fonctions suivantes, considérez une solution de sauvegarde d'entreprise :
  - Restauration fichier par fichier
  - Vérification des médias de sauvegarde
  - Gestion des médias
- **Instantanés de systèmes de fichiers et restauration d'instantanés** : exécutez les commandes `zfs snapshot` et `zfs rollback` pour créer facilement une copie d'un système de fichiers et restaurer une version précédente d'un système de fichiers, le cas échéant. Par exemple, vous pouvez utiliser cette solution pour restaurer un ou plusieurs fichiers issus d'une version précédente d'un système de fichiers.

  Pour de plus amples informations sur la création et la restauration d'instantané, reportez-vous à la section "Présentation des instantanés ZFS" à la page 219.

- **Enregistrement d'instantanés** : utilisez les commandes `zfs send` et `zfs receive` pour envoyer et recevoir un instantané ZFS. Vous pouvez enregistrer les modifications incrémentielles entre instantanés, mais la restauration individuelle de fichiers est impossible. L'instantané du système doit être restauré dans son intégralité. Ces commandes ne constituent pas une solution de sauvegarde complète pour l'enregistrement de vos données ZFS.
- **Réplication distante** : utilisez les commandes `zfs send` et `zfs receive` pour copier un système de fichiers d'un système vers un autre. Ce processus diffère d'un produit de gestion de volume classique qui pourrait mettre les périphériques en miroir dans un WAN. Aucune configuration ni aucun matériel spécifique n'est requis. La réplication de systèmes de fichiers ZFS a ceci d'avantageux qu'elle permet de recréer un système de fichiers dans un pool de stockage et de spécifier différents niveaux de configuration pour le nouveau pool, comme RAID-Z, mais avec des données de système de fichiers identiques.
- **Utilitaires d'archivage** : enregistrez les données ZFS à l'aide d'utilitaires d'archivage tels que `tar`, `cpio` et `pax`, ou des produits de sauvegarde tiers. Actuellement, les deux utilitaires `tar` et `cpio` traduisent correctement les ACL de type NFSv4, contrairement à l'utilitaire `pax`.

Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde

Outre les commandes `zfs send` et `zfs receive`, vous pouvez utiliser des utilitaires d'archivage, tels que les commandes `tar` et `cpio` pour enregistrer des fichiers ZFS. Ces utilitaires enregistrent et restaurent les attributs de fichiers et les ACL ZFS. Vérifiez les options adéquates des commandes `tar` et `cpio`.

Pour des informations les plus récentes sur les problèmes relatifs aux produits de sauvegarde ZFS et tiers, reportez-vous aux notes de version de Solaris 10.

Reconnaissance des flux d'instantané ZFS

Un instantané d'un système de fichiers ou volume ZFS est converti en flux d'instantané à l'aide de la commande `zfs send`. Ensuite, vous pouvez utiliser le flux d'instantané pour recréer un système de fichiers ou volume ZFS à l'aide de la commande `zfs receive`.

Selon les options `zfs send` utilisées pour créer le flux d'instantané, différents formats de flux sont générés.

- Flux complet : se compose du contenu intégral du jeu de données, depuis sa création jusqu'à la prise de l'instantané spécifié.

  Le flux par défaut généré par la commande `zfs send` est un flux complet. Il contient un système de fichiers ou un volume, jusqu'à et y compris l'instantané spécifié. Le flux ne contient pas d'autre instantané que celui spécifié sur la ligne de commande.

- Flux incrémentiel : se compose des différences entre deux instantanés.

Un package de flux est un type de flux contenant un ou plusieurs flux incrémentiels. Il existe trois types de packages de flux :

- Package de flux de réplication : se compose du jeu de données spécifié et de ses descendants. Il inclut tous les instantanés intermédiaires. Si l'origine d'un jeu de données cloné n'est pas un descendant de l'instantané spécifié sur la ligne de commande, le jeu de données d'origine n'est pas inclus dans le package de flux. Pour recevoir le flux, le jeu de données d'origine doit exister dans le pool de stockage de destination.

  Examinez la liste de jeux de données suivis de leur origine suivante. Nous supposerons qu'ils ont été créés dans l'ordre dans lequel ils apparaissent ci-dessous :

```
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           -
```

  Un package de flux de réplication créé en respectant la syntaxe suivante :

```
# zfs send -R pool/b@send ....
```

  Se compose des flux complets et incrémentiels suivants :

```
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
```

  Dans la sortie qui précède, l'instantané `pool/a/1@clone` n'est pas inclus dans le package de flux de réplication. En l'état, ce package de flux de réplication peut uniquement être reçu dans un pool possédant déjà l'instantané `pool/a/1@clone` .

- Package de flux récursif : se compose du jeu de données spécifié et de ses descendants. A la différence des packages de flux de réplication, les instantanés intermédiaires ne sont pas inclus, sauf s'ils constituent l'origine d'un jeu de données cloné inclus dans le flux. Par défaut, si l'origine d'un jeu de données n'est pas un descendant de l'instantané spécifié sur la ligne de commande, le comportement est le même que pour les flux de réplication. Néanmoins, un flux récursif autonome, comme décrit ci-après, est créé de manière à ce qu'il n'y ait aucune dépendance externe.

Un package de flux récursif créé en respectant la syntaxe suivante :

```
# zfs send -r pool/b@send ...
```

Se compose des flux complets et incrémentiels suivants :

```
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
```

Dans la sortie qui précède, l'instantané `pool/a/1@clone` n'est pas inclus dans le package de flux récursif. En l'état, ce package de flux récursif peut uniquement être reçu dans un pool qui possède déjà l'instantané `pool/a/1@clone` . Ce comportement est similaire au scénario du package de flux de réplication décrit plus haut.

- Package de flux récursif autonome : ne dépend d'aucun jeu de données non inclus dans le package de flux. Le package de flux récursif créé en respectant la syntaxe suivante :

  ```
  # zfs send -rc pool/b@send ...
  ```

  Se compose des flux complets et incrémentiels suivants :

  ```
  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
  ```

  Notez que le flux récursif autonome possède un flux complet de l'instantané `pool/b/1@clone2`, ce qui permet la réception de l'instantané `pool/b/1` sans dépendance externe.

Envoi d'un instantané ZFS

Vous pouvez utiliser la commande `zfs send` pour envoyer une copie d'un flux d'instantané et recevoir ce flux dans un autre pool du même système ou dans un autre pool d'un système différent utilisé pour stocker les données de sauvegarde. Par exemple, pour envoyer le flux d'instantané à un pool différent du même système, employez une syntaxe du type suivant :

```
# zfs send tank/dana@snap1 | zfs recv spool/ds01
```

Vous pouvez utiliser `zfs recv` en tant qu'alias pour la commande `zfs receive`.

Si vous envoyez le flux de l'instantané à un système différent, envoyez la sortie de la commande `zfs send` à la commande `ssh`. Par exemple :

```
sys1# zfs send tank/dana@snap1 | ssh sys2 zfs recv newtank/dana
```

Lors de l'envoi d'un flux complet, le système de fichiers de destination ne doit pas exister.

Vous pouvez envoyer les données incrémentielles à l'aide de l'option `zfs send -i`. Par exemple :

```
sys1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

Le premier argument (`snap1`) correspond à l'instantané le plus ancien, le second (`snap2`) à l'instantané le plus récent. Dans ce cas, le système de fichiers `newtank/dana` doit déjà exister pour que la réception incrémentielle s'effectue correctement.

Remarque – L'accès à des informations de fichier dans le système de fichiers reçu à l'origine, peut causer l'échec d'une opération de réception d'instantané incrémentale avec un message similaire à celui-ci :

```
cannot receive incremental stream of tank/dana@snap2 into newtank/dana:
most recent snapshot of tank/dana@snap2 does not match incremental source
```

Envisagez de définir la propriété `atime` sur off si vous avez besoin d'accéder à des informations de fichier dans le système de fichiers reçu à l'origine et si vous avez aussi besoin de recevoir des instantanés incrémentaux dans le système de fichiers reçu.

La source du *snap1* incrémentiel peut être spécifiée comme étant le dernier composant du nom de l'instantané. Grâce à ce raccourci, il suffit de spécifier le nom après le signe `@` pour le *snap1*, qui est considéré comme provenant du même système de fichiers que le *snap2*. Par exemple :

```
sys1# zfs send -i snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

Cette syntaxe de raccourci est équivalente à la syntaxe incrémentielle de l'exemple précédent.

Le message s'affiche en cas de tentative de génération d'un flux incrémentiel à partir d'un `instantané1` provenant d'un autre système de fichiers :

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Si vous devez stocker de nombreuses copies, envisagez de compresser une représentation de flux d'instantané ZFS à l'aide de la commande `gzip`. Par exemple :

```
# zfs send pool/fs@snap | gzip > backupfile.gz
```

Réception d'un instantané ZFS

Gardez les points suivants à l'esprit lorsque vous recevez un instantané d'un système de fichiers :

- L'instantané et le système de fichiers sont reçus.
- Le système de fichiers et tous les systèmes de fichiers descendants sont démontés.
- Les systèmes de fichiers sont inaccessibles tant qu'ils sont en cours de réception.
- Le système de fichiers d'origine à recevoir ne doit pas exister tant qu'il est en cours de transfert.

- Si ce nom existe déjà, vous pouvez utiliser la commande `zfs rename` pour renommer le système de fichiers.

Par exemple :

```
# zfs send tank/gozer@0830 > /bkups/gozer.083006
# zfs receive tank/gozer2@today < /bkups/gozer.083006
# zfs rename tank/gozer tank/gozer.old
# zfs rename tank/gozer2 tank/gozer
```

Si vous apportez des modifications au système de fichiers de destination et souhaitez effectuer un autre envoi incrémentiel d'instantané, vous devez au préalable restaurer le système de fichiers destinataire.

Voyez l'exemple suivant. Modifiez tout d'abord le système de fichiers comme suit :

```
sys2# rm newtank/dana/file.1
```

Effectuez ensuite un envoi incrémentiel de `char/dana@snap3`. Cependant, vous devez d'abord annuler (roll back) le système de fichiers destinataire pour permettre la réception du nouvel instantané incrémentiel. Vous pouvez aussi utiliser l'option `-F` pour éviter l'étape de restauration. Par exemple :

```
sys1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh sys2 zfs recv -F newtank/dana
```

Lors de la réception d'un instantané incrémentiel, le système de fichiers de destination doit déjà exister.

Si vous apportez des modifications au système de fichiers sans restaurer le système de fichiers destinataire pour permettre la réception du nouvel instantané incrémentiel, ou si vous ne spécifiez pas l'option `-F`, un message similaire au message suivant s'affiche :

```
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
```

Les vérifications suivantes sont requises pour assurer l'exécution de l'option `-F` :

- Si l'instantané le plus récent ne correspond pas à la source incrémentielle, la restauration et la réception ne s'effectuent pas intégralement et un message d'erreur s'affiche.
- Si vous avez fourni accidentellement le nom d'un système de fichiers qui ne correspond pas à la source incrémentielle dans la commande `zfs receive`, la restauration et la réception ne s'effectuent pas correctement et le message d'erreur suivant s'affiche :

  ```
  cannot send 'pool/fs@name': not an earlier snapshot from the same fs
  ```

Application de différentes valeurs de propriété à un flux d'instantané ZFS

Vous pouvez envoyer un flux d'instantané ZFS avec une certaine valeur de propriété de système de fichiers, mais vous pouvez spécifier une valeur de propriété locale différente lorsque le flux de l'instantané est reçu. Vous pouvez également indiquer que la valeur de propriété d'origine doit être utilisée lorsque le flux d'instantané est reçu pour recréer le système de fichiers d'origine. En outre, vous pouvez désactiver une propriété de système de fichiers lorsque le flux d'instantané est reçu.

- Utilisez la commande `zfs inherit -S` pour rétablir la valeurs de propriété locale reçue, le cas échéant. Si une propriété ne reçoit aucune valeur, le comportement de la commande `zfs inherit -S` est le même que la commande `zfs inherit` sans l'option `-S`. Si la propriété ne reçoit aucune valeur, la commande `zfs inherit` masque la valeur reçue par la valeur héritée jusqu'à ce que l'émission d'une commande `zfs inherit -S` rétablisse la valeur reçue.
- Vous pouvez utiliser la commande `zfs get -o` pour prendre en compte la nouvelle colonne `RECEIVED` ajoutée. Vous pouvez également utiliser la commande `zfs get -o all` pour ajouter toutes les colonnes, y compris la colonne `RECEIVED`.
- Vous pouvez utiliser l'option `zfs send -p` pour ajouter des propriétés dans le flux d'envoi sans l'option `-R`.
- L'option `zfs receive -e` permet d'utiliser le dernier élément du nom de l'instantané envoyé pour définir le nom du nouvel instantané. L'exemple suivant envoie l'instantané `poola/bee/cee@1` au système de fichiers `poold/eee` et utilise uniquement le dernier élément (`cee@1`) du nom de l'instantané pour créer le système de fichiers et l'instantané reçus.

```
# 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  -
```

Dans certains cas, les propriétés du système de fichiers dans un flux envoyé ne peuvent pas s'appliquer au système de fichiers récepteur ou aux propriétés du système de fichiers local, comme la valeur de propriété `mountpoint`, et risquent d'interférer avec une restauration.

Par exemple, dans le système de fichiers `tank/données`, la propriété `compression` est désactivée. Un instantané du système de fichiers `tank/data` est envoyé avec des propriétés (option `-p`) à un pool de sauvegarde et est reçu avec la propriété `compression` activée.

```
# 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
```

Dans l'exemple, la propriété `compression` est activée lorsque l'instantané est reçu dans `bpool`. Par conséquent, pour `bpool/data`, la valeur `compression` est activée.

Si ce flux d'instantané est envoyé à un nouveau pool, `restorepool`, à des fins de récupération, vous pouvez être amené à conserver toutes les propriétés de l'instantané d'origine. Dans ce cas, vous devez utiliser la commande `zfs send -b` pour restaurer les propriétés de l'instantané d'origine. Par exemple :

```
# 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
```

Dans l'exemple, la valeur de compression est `off`, elle représente la valeur de compression de l'instantané du système de fichiers `tank/data` d'origine.

Si vous disposez d'une valeur de propriété de système de fichiers local dans un flux d'instantané et que vous souhaitez désactiver la propriété lors de sa réception, utilisez la commande `zfs receive -x`. Par exemple, la commande suivante envoie un flux d'instantané récursif des systèmes de fichiers du répertoire personnel avec toutes les propriétés de système de fichiers réservées à un pool de sauvegarde, mais sans les valeurs de propriété du quota.

```
# 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     -      -
```

Si l'instantané récursif n'a pas été reçu avec l'option `-x`, la propriété de quota doit être définie dans les systèmes de fichiers reçus.

```
# 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     -      -
```

Envoi et réception de flux d'instantanés ZFS complexes

Cette section décrit l'utilisation des options `zfs send -I` et `-R` pour envoyer et recevoir des flux d'instantanés plus complexes.

Gardez les points suivants à l'esprit lors de l'envoi et de la réception de flux d'instantanés ZFS complexes :

- Utilisez l'option `zfs send -I` pour envoyer tous les flux incrémentiels d'un instantané à un instantané cumulé. Vous pouvez également utiliser cette option pour envoyer un flux incrémentiel de l'instantané d'origine pour créer un clone. L'instantané d'origine doit déjà exister sur le côté récepteur afin d'accepter le flux incrémentiel.
- Utilisez l'option `zfs send -R` pour envoyer un flux de réplication de tous les systèmes de fichiers descendants. Une fois le flux de réplication reçu, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.
- Lorsque l'option `zfs send -r` est utilisée sans l'option `-c` et lorsque l'option `zfs send -R` est utilisée, les packages de flux omettent dans certains cas l'`origin` des clones. Pour plus d'informations, reportez-vous à la section "Reconnaissance des flux d'instantané ZFS" à la page 230.
- Vous pouvez utiliser les deux options pour envoyer un flux de réplication incrémentiel.
  - Les modifications des propriétés sont conservées, tout comme les opérations `rename` et `destroy` des instantanés et des systèmes de fichiers.
  - Si l'option `zfs recv -F` n'est pas spécifiée lors de la réception du flux de réplication, les opérations `destroy` du jeu de données sont ignorées. La syntaxe de `zfs recv -F` dans ce cas peut conserve également sa signification de *récupération le cas échéant*.
  - Tout comme dans les autres cas `-i` ou `-I` (autres que `zfs send -R`), si l'option `-I` est utilisée, tous les instantanés créés entre `snapA` et `snapD` sont envoyés. Si l'option `-i` est utilisée, seul `snapD` (pour tous les descendants) est envoyé.
- Pour recevoir ces nouveaux types de flux `zfs send`, le système récepteur doit exécuter une version du logiciel capable de les envoyer. La version des flux est incrémentée.

  Vous pouvez cependant accéder à des flux d'anciennes versions de pool en utilisant une version plus récente du logiciel. Vous pouvez par exemple envoyer et recevoir des flux créés à l'aide des nouvelles options à partir d'un pool de la version 3. Vous devez par contre exécuter un logiciel récent pour recevoir un flux envoyé avec les nouvelles options.

EXEMPLE 6–1 Envoi et réception de flux d'instantanés ZFS complexes

Plusieurs instantanés incrémentiels peuvent être regroupés en un seul instantané à l'aide de l'option `zfs send -I`. Par exemple :

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

EXEMPLE 6–1 Envoi et réception de flux d'instantanés ZFS complexes *(Suite)*

Vous pouvez ensuite supprimer `snapB`, `snapC` et `snapD`.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Pour recevoir les instantanés combinés, vous devez utiliser la commande suivante :

```
# 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  -
```

Vous pouvez également utiliser la commande `zfs send -I` pour regrouper un instantané et un clone d'instantané en un nouveau jeu de données. Par exemple :

```
# 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
```

Vous pouvez utiliser la commande `zfs send -R` pour répliquer un système de fichiers ZFS et tous ses systèmes de fichiers descendants, jusqu'à l'instantané nommé. Une fois ce flux reçu, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.

Dans l'exemple suivant, des instantanés des systèmes de fichiers utilisateur sont créés. Un flux de réplication de tous les instantanés utilisateur est créé. Les systèmes de fichiers et instantanés d'origine sont ensuite détruits et récupérés.

```
# 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
```

EXEMPLE 6–1 Envoi et réception de flux d'instantanés ZFS complexes *(Suite)*

```
# 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  -
```

Dans l'exemple suivant, la commande `zfs send -R` a été utilisée pour répliquer le système de fichiers `users` et ses descendants et pour envoyer le flux répliqué vers un autre pool, `users2`.

```
# zfs create users2 mirror c0t1d0 c1t1d0
# zfs receive -F -d users2 < /snaps/users-R
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
users                 224K  33.2G    22K  /users
users@today              0      -    22K  -
users/user1           33K  33.2G    18K  /users/user1
users/user1@today     15K      -    18K  -
users/user2           18K  33.2G    18K  /users/user2
users/user2@today        0      -    18K  -
users/user3           18K  33.2G    18K  /users/user3
users/user3@today        0      -    18K  -
users2                188K  16.5G    22K  /users2
users2@today             0      -    22K  -
users2/user1          18K  16.5G    18K  /users2/user1
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  -
```

Réplication distante de données ZFS

Les commandes `zfs send` et `zfs recv` permettent d'effectuer une copie distante d'une représentation de flux d'instantané d'un système vers un autre. Par exemple :

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Cette commande envoie les données de l'instantané `tank/cindy@today` et les reçoit dans le système de fichiers `sandbox/restfs`. La commande suivante crée également un instantané `restfs@aujourd'hui` sur le système `newsys`. Dans cet exemple, l'utilisateur a été configuré pour utiliser `ssh` dans le système distant.

◆ ◆ ◆ CHAPITRE 7

Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS

Ce chapitre décrit l'utilisation des listes de contrôle d'accès (ACL, Access Control List) pour protéger les fichiers ZFS en accordant des autorisations à un niveau de granularité plus fin que les autorisations UNIX standard.

Ce chapitre contient les sections suivantes :

- "Modèle ACL Solaris" à la page 241
- "Configuration d'ACL dans des fichiers ZFS" à la page 248
- "Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé" à la page 251
- "Configuration et affichage d'ACL dans des fichiers ZFS en format compact" à la page 261

Modèle ACL Solaris

Les versions précédentes de Solaris assuraient la prise en charge d'une implémentation ACL reposant principalement sur la spécification POSIX-draft ACL. Les ACL basées sur POSIX-draft sont utilisées pour protéger les fichiers UFS et sont traduites par les versions de NFS antérieures à NFSv4.

Grâce à l'introduction de NFSv4, un nouveau modèle d'ACL assure entièrement la prise en charge de l'interopérabilité qu'offre NFSv4 entre les clients UNIX et non UNIX. La nouvelle implémentation d'ACL, telle que définie dans les spécifications NFSv4, fournit des sémantiques bien plus riches, basées sur des ACL NT.

Les différences principales du nouveau modèle d'ACL sont les suivantes :

- Modèle basé sur la spécification NFSv4 et similaire aux ACL de type NT.
- Jeu de privilèges d'accès bien plus granulaire. Pour plus d'informations, reportez-vous au Tableau 7–2.
- Configuration et affichage avec les commandes `chmod` et `ls`, et non les commandes `setfacl` et `getfacl`.

- Sémantique d'héritage bien plus riche pour déterminer comment les privilèges d'accès sont appliqués d'un répertoire à un sous-répertoire, et ainsi de suite. Pour plus d'informations, reportez-vous à la section "Héritage d'ACL" à la page 246.

Les deux modèles d'ACL assurent un contrôle d'accès à un niveau de granularité plus fin que celui disponible avec les autorisations de fichier standard. De façon similaire aux listes de contrôle d'accès POSIX-draft, les nouvelles ACL se composent de plusieurs ACE (Access Control Entry, entrées de contrôle d'accès).

Les ACL POSIX-draft utilisent une seule entrée pour définir quelles autorisations sont accordées et lesquelles sont refusées. Le nouveau modèle d'ACL dispose de deux types d'ACE qui affectent la vérification d'accès : `ALLOW` et `DENY`. Il est en soi impossible de déduire de toute entrée de contrôle d'accès (ACE) définissant un groupe d'autorisations si les autorisations qui n'ont pas été définies dans cette ACE sont ou non accordées.

La conversion entre les ACL NFSv4 et les ACL POSIX-draft s'effectue comme suit :

- Si vous employez un utilitaire compatible avec les ACL (les commandes `cp`, `mv`, `tar`, `cpio` ou `rcp`, par exemple) pour transférer des fichiers UFS avec des ACL vers un système de fichiers ZFS, les ACL POSIX-draft sont converties en ACL NFSv4 équivalentes.
- Les ACL NFSv4 sont converties en ACL POSIX-draft. Un message tel que le suivant s'affiche si une ACL NFSv4 n'est pas convertie en ACL POSIX-draft :

  ```
  # cp -p filea /var/tmp
  cp: failed to set acl entries on /var/tmp/filea
  ```
- Si vous créez une archive `cpio` ou `tar` UFS avec l'option de conservation des ACL (`tar -p` ou `cpio -P`) dans un système exécutant la version actuelle de Solaris, les ACL sont perdues en cas d'extraction de l'archive sur un système exécutant une version précédente de Solaris.

  Tous les fichiers sont extraits avec les modes de fichier corrects, mais les entrées d'ACL sont ignorées.
- Vous pouvez utiliser la commande `ufsrestore` pour restaurer des données dans un système de fichiers ZFS. Si les données d'origine incluent des ACL POSIX-style, elles sont converties en ACL NFSv4-style.
- En cas de tentative de configuration d'une ACL SFSv4 dans un fichier UFS, un message tel que le suivant s'affiche :

  ```
  chmod: ERROR: ACL type's are different
  ```
- En cas de tentative de configuration d'une ACL POSIX dans un fichier ZFS, un message tel que le suivant s'affiche :

  ```
  # getfacl filea
  File system doesn't support aclent_t style ACL's.
  See acl(5) for more information on Solaris ACL support.
  ```

Pour obtenir des informations sur les autres limitations des ACL et des produits de sauvegarde, reportez-vous à la section "Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde" à la page 230.

Descriptions de syntaxe pour la configuration des ACL

Deux formats d'ACL de base sont fournis comme suit :

- **ACL triviales** – Contient uniquement des entrées `user`, `group` et `owner` UNIX traditionnelles.
- **ACL non triviale** – Contient plus d'entrées qu'owner, group et everyone, inclut des indicateurs d'héritage ou les entrées sont triées d'une manière non traditionnelle.

Syntaxe pour la configuration d'ACL triviales

`chmod [options] A[index]{+|=}owner@ |group@ |everyone@:`
autorisations-d'accès`/...[:`*indicateurs-héritage*`]: deny | allow` *file*

`chmod [options] A-owner@, group@, everyone@:` *autorisations-d'accès*
`/...[:`*indicateurs-héritage*`]:deny | allow` *file ...*

`chmod [options] A[index]-` *file*

Syntaxe pour la configuration d'ACL non triviales

`chmod [options] A[index]{+|=}user|group:name:` *autorisations-d'accès*
`/...[:`*indicateurs-héritage*`]:deny | allow` *file*

`chmod [options] A-user|group:name:` *autorisations-d'accès* `/...[:`*indicateurs-héritage*`]:deny | allow` *file ...*

`chmod [options] A[index]-` *file*

owner@, group@, everyone@
: Identifie le *type d'entrée d'ACL* pour la syntaxe d'ACL triviale. Pour obtenir une description des *types d'entrées d'ACL*, reportez-vous au Tableau 7–1.

utilisateur ou groupe :*ID-entrée-ACL=nom utilisateur ou nom groupe*
: Identifie le *type d'entrée d'ACL* pour la syntaxe d'ACL explicite. Le *type d'entrée d'ACL* pour l'utilisateur et le groupe doit également contenir l'*ID d'entrée d'ACL*, le *nom d'utilisateur* ou le *nom de groupe*. Pour obtenir une description des *types d'entrées d'ACL*, reportez-vous au Tableau 7–1.

autorisations-d'accès`/.../`
: Identifie les autorisations d'accès accordées ou refusées. Pour obtenir une description des privilèges d'accès d'ACL, reportez-vous au Tableau 7–2.

indicateurs-héritage
: Identifie une liste optionnelle d'indicateurs d'héritage d'ACL. Pour une description des indicateurs d'héritage d'ACL, reportez-vous au Tableau 7–4.

`deny` | `allow`
: Détermine si les autorisations d'accès sont accordées ou refusées.

Dans l'exemple suivant, aucune valeur d'*ID d'entrée d'ACL* n'existe pour owner@, group@ ou everyone@.

```
group@:write_data/append_data/execute:deny
```

L'exemple suivant inclut un *ID d'entrée d'ACL* car un utilisateur spécifique (*type d'entrée d'ACL*) est inclus dans la liste.

```
0:user:gozer:list_directory/read_data/execute:allow
```

Lorsqu'une entrée d'ACL s'affiche, elle est similaire à celle-ci :

```
2:group@:write_data/append_data/execute:deny
```

La désignation **2** ou *ID d'index* dans cet exemple identifie l'entrée d'ACL dans la plus grande ACL, qui peut présenter plusieurs entrées pour le propriétaire, des UID spécifiques, un groupe et pour tous. Vous pouvez spécifier l'*ID d'index* avec la commande `chmod` pour identifier la partie de l'ACL que vous souhaitez modifier. Par exemple, vous pouvez identifier l'ID d'index 3 par `A3` dans la commande `chmod` comme ci-dessous :

```
chmod A3=user:venkman:read_acl:allow filename
```

Les types d'entrées d'ACL (qui sont les représentations d'ACL du propriétaire, du groupe et autres) sont décrits dans le tableau suivant.

TABLEAU 7–1 Types d'entrées d'ACL

Type d'entrée d'ACL	Description
`owner@`	Spécifie l'accès accordé au propriétaire de l'objet.
`group@`	Spécifie l'accès accordé au groupe propriétaire de l'objet.
`everyone@`	Spécifie l'accès accordé à tout utilisateur ou groupe ne correspondant à aucune autre entrée d'ACL.
`user`	Avec un nom d'utilisateur, spécifie l'accès accordé à un utilisateur supplémentaire de l'objet. Doit inclure l'*ID d'entrée d'ACL* qui contient un *username* ou un *userID*. Le type d'entrée d'ACL est incorrect si la valeur n'est ni un UID numérique, ni un *username*.
`group`	Avec un nom de groupe, spécifie l'accès accordé à un utilisateur supplémentaire de l'objet. Doit inclure l'*ID d'entrée d'ACL* qui contient un *groupname* ou un *groupID*. Le type d'entrée d'ACL est incorrect si la valeur n'est ni un GID numérique, ni un *groupname*.

Les privilèges d'accès sont décrits dans le tableau suivant.

TABLEAU 7–2 Privilèges d'accès d'ACL

Privilège d'accès	Privilège d'accès compact	Description
`add_file`	`w`	Autorisation d'ajouter un fichier à un répertoire.
`add_subdirectory`	`p`	Dans un répertoire, autorisation de créer un sous-répertoire.
`append_data`	`p`	Non implémentée actuellement.
`delete`	`d`	Autorisation de supprimer un fichier. Pour plus d'informations sur le comportement spécifique de l'autorisation `delete`, reportez-vous au Tableau 7–3.
`delete_child`	`D`	Autorisation de supprimer un fichier ou un répertoire au sein d'un répertoire. Pour plus d'informations sur le comportement spécifique de l'autorisation `delete_child` , reportez-vous au Tableau 7–3.
`execute`	`x`	Autorisation d'exécuter un fichier ou d'effectuer une recherche dans le contenu d'un répertoire.
`list_directory`	`r`	Autorisation de dresser la liste du contenu d'un répertoire.
`read_acl`	`c`	Autorisation de lire l'ACL (`ls`).
`read_attributes`	`a`	Autorisation de lire les attributs de base (non ACL) d'un fichier. Considérez les attributs de base comme les attributs de niveau stat. L'autorisation de ce bit de masque d'accès signifie que l'entité peut exécuter `ls`(1) et `stat`(2).
`read_data`	`r`	Autorisation de lire le contenu du fichier.
`read_xattr`	`R`	Autorisation de lire les attributs étendus d'un fichier ou d'effectuer une recherche dans le répertoire d'attributs étendus d'un fichier.
`synchronize`	`s`	Non implémentée actuellement.
`write_xattr`	`W`	Autorisation de créer des attributs étendus ou d'écrire dans le répertoire d'attributs étendus. L'attribution de cette autorisation à un utilisateur signifie que ce dernier peut créer un répertoire d'attributs étendus pour un fichier. Les autorisations du fichier d'attributs contrôlent l'accès de l'utilisateur à l'attribut.
`write_data`	`w`	Autorisation de modifier ou de remplacer le contenu d'un fichier.
`write_attributes`	`A`	Autorisation de remplacer les durées associées à un fichier ou un répertoire par une valeur arbitraire.
`write_acl`	`C`	Autorisation d'écriture sur l'ACL ou capacité de la modifier à l'aide de la commande `chmod`.

TABLEAU 7–2 Privilèges d'accès d'ACL *(Suite)*

Privilège d'accès	Privilège d'accès compact	Description
`write_owner`	`o`	Autorisation de modifier le propriétaire ou le groupe d'un fichier. Ou capacité d'exécuter les commandes `chown` ou `chgrp` sur le fichier. Autorisation de devenir propriétaire d'un fichier ou autorisation de définir la propriété de groupe du fichier sur un groupe dont fait partie l'utilisateur. Le privilège `PRIV_FILE_CHOWN` est requis pour définir la propriété de fichier ou de groupe sur un groupe ou un utilisateur arbitraire.

Le tableau suivant fournit des détails supplémentaires sur les comportements `delete` et `delete_child` d'ACL.

TABLEAU 7–3 Comportement des autorisations delete et delete_child d'ACL.

Droits d'accès au répertoire parent	Autorisations d'objet cible		
	L'ACL autorise la suppression	L'ACL refuse la suppression	Autorisation de suppression non spécifiée
L'ACL autorise `delete_child`	Autorisation	Autorisation	Autorisation
L'ACL refuse `delete_child`	Autorisation	Refus	Refus
L'ACL autorise uniquement `write` et `execute`	Autorisation	Autorisation	Autorisation
L'ACL refuse `write` et `execute`	Autorisation	Refus	Refus

Héritage d'ACL

L'héritage d'ACL a pour finalité de permettre à un fichier ou répertoire récemment créé d'hériter des ACL qui leur sont destinées, tout en tenant compte des bits d'autorisation existants dans le répertoire parent.

Par défaut, les ACL ne sont pas propagées. Si vous configurez une ACL non triviale dans un répertoire, aucun répertoire enfant n'en hérite. Vous devez spécifier l'héritage d'une ACL dans un fichier ou un répertoire.

Les indicateurs d'héritage facultatifs sont décrits dans le tableau suivant.

TABLEAU 7–4 Indicateurs d'héritage d'ACL

Indicateur d'héritage	Indicateur d'héritage compact	Description
`file_inherit`	`f`	Hérite de l'ACL uniquement à partir du répertoire parent vers les fichiers du répertoire.
`dir_inherit`	`d`	Hérite de l'ACL uniquement à partir du répertoire parent vers les sous-répertoires du répertoire.
`inherit_only`	`i`	Hérite de l'ACL à partir du répertoire parent mais ne s'applique qu'aux fichiers et sous-répertoires récemment créés, pas au répertoire lui-même. Cet indicateur requiert les indicateurs `file_inherit` et/ou `dir_inherit` afin de spécifier ce qui doit être hérité.
`no_propagate`	`n`	N'hérite que de l'ACL provenant du répertoire parent vers le contenu de premier niveau du répertoire, et non les contenus de second niveau et suivants. Cet indicateur requiert les indicateurs `file_inherit` et/ou `dir_inherit` afin de spécifier ce qui doit être hérité.
-	SO	Aucune autorisation n'est accordée.

De plus, vous pouvez configurer une stratégie d'héritage d'ACL par défaut plus ou moins stricte sur le système de fichiers à l'aide de la propriété de système de fichiers `aclinherit`. Pour de plus amples informations, consultez la section suivante.

Propriétés ACL

Le système de fichiers ZFS inclut les propriétés d'ACL suivantes permettant de déterminer le comportement spécifique de l'héritage d'ACL et des interactions d'ACL avec les opérations `chmod`.

- `aclinherit` : détermine le comportement d'héritage d'ACL. Les valeurs possibles sont les suivantes :
  - `discard` : pour les nouveaux objets, aucune entrée d'ACL n'est héritée lors de la création d'un fichier ou d'un répertoire. L'ACL dans le fichier ou le répertoire est égale au mode d'autorisation du fichier ou répertoire.
  - `noallow` : pour les nouveaux objets, seules les entrées d'ACL héritables dont le type d'accès est `deny` sont héritées.
  - `restricted` : pour les nouveaux objets, les autorisations `write_owner` et `write_acl` sont supprimées lorsqu'une entrée d'ACL est héritée.

- `passthrough` : lorsqu'une valeur de propriété est définie sur `passthrough`, les fichiers sont créés dans un mode déterminé par les ACE héritées. Si aucune ACE pouvant être héritée n'affecte le mode, ce mode est alors défini en fonction du mode demandé à partir de l'application.
- `passthrough-x` : a la même sémantique que `passthrough`, si ce n'est que lorsque `passthrough-x` est activé, les fichiers sont créés avec l'autorisation d'exécution (`x`), mais uniquement si l'autorisation d'exécution est définie en mode de création de fichier et dans une entrée de contrôle d'accès (ACE) pouvant être héritée et qui affecte le mode.

Le mode par défaut de `aclinherit` est `restricted`.

- `aclmode` – Modifie le comportement des ACL lorsqu'un fichier est créé et contrôle la modification des ACL au cours d'une opération `chmod`. Les valeurs possibles sont les suivantes :
  - `discard` :un système de fichiers dont la valeur de la propriété `aclmode` est `discard` supprime toutes les entrées d'ACL qui ne représentent pas le mode du fichier. Il s'agit de la valeur par défaut.
  - `mask` : un système de fichiers dont la valeur de la propriété `aclmode` est `mask` restreint les autorisations utilisateur ou groupe. Les autorisations sont réduites de manière à ne pas excéder les bits d'autorisation du groupe, à moins qu'il ne s'agisse d'une entrée utilisateur possédant le même UID que le propriétaire du fichier ou du répertoire. Dans ce cas, les autorisations d'ACL sont réduites de manière à ne pas excéder les bits d'autorisation du propriétaire. La valeur de masque préserve en outre l'ACL lors des modifications de mode successives, à condition qu'aucune opération de jeu d'ACL explicite n'ait été effectuée.
  - `passthrough` : un système de fichiers avec une propriété `aclmode` de `passthrough` indique qu'aucune modification n'est apportée à l'ACL en dehors de la génération des entrées d'ACL nécessaires pour représenter le nouveau mode du fichier ou du répertoire.

  Le mode par défaut pour `aclmode` est `discard`.

  Pour plus d'informations sur l'utilisation de la propriété `aclmode`, reportez-vous à l'Exemple 7–13.

Configuration d'ACL dans des fichiers ZFS

Dans la mesure où elles sont implémentées avec ZFS, les ACL se composent d'un tableau d'entrées d'ACL. ZFS fournit un modèle d'ACL *pur*, dans lequel tous les fichiers présentent une ACL. En règle générale, cette liste est *triviale* dans la mesure où elle ne représente que les entrées `propriétaire/groupe/autre` UNIX classiques.

Les fichiers ZFS disposent toujours de bits d'autorisation et d'un mode, mais ces valeurs constituent plutôt un cache de ce que représente une ACL. Par conséquent, si vous modifiez les autorisations du fichier, son ACL est mise à jour en conséquence. En outre, si vous supprimez

une ACL non triviale qui accordait à un utilisateur l'accès à un fichier ou à un répertoire, il est possible que cet utilisateur y ait toujours accès en raison des bits d'autorisation qui accordent l'accès à un groupe ou à tous les utilisateurs. L'ensemble des décisions de contrôle d'accès est régi par les autorisations représentées dans l'ACL d'un fichier ou d'un répertoire.

Les règles principales d'accès aux ACL dans un fichier ZFS sont comme suit :

- ZFS traite les entrées d'ACL dans l'ordre dans lesquelles elles sont répertoriées dans l'ACL, en partant du haut.
- Seules les entrées d'ACL disposant d'un " who " correspondant au demandeur d'accès sont traitées.
- Une fois l'autorisation allow accordée, cette dernière ne peut plus être refusée par la suite par une entrée d'ACL de refus dans le même jeu de d'autorisations d'ACL.
- Le propriétaire du fichier dispose de l'autorisation `write_acl` de façon inconditionnelle, même si celle-ci est explicitement refusée. Dans le cas contraire, toute autorisation non spécifiée est refusée.

  Dans les cas d'autorisations deny ou lorsqu'une autorisation d'accès est manquante, le sous-système de privilèges détermine la demande d'accès accordée pour le propriétaire du fichier ou pour le superutilisateur. Ce mécanisme évite que les propriétaires de fichiers puissent accéder à leurs fichiers et permet aux superutilisateurs de modifier les fichiers à des fins de récupération.

Si vous configurez une ACL non triviale dans un répertoire, les enfants du répertoire n'en héritent pas automatiquement. Si vous configurez une ACL non triviale, et souhaitez qu'elle soit héritée par les enfants du répertoire, vous devez utiliser les indicateurs d'héritage d'ACL. Pour plus d'informations, reportez-vous au Tableau 7–4 et à la section “Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé” à la page 255.

Lorsque vous créez un fichier, en fonction de la valeur `umask`, une ACL triviale par défaut, similaire à la suivante, est appliquée :

```
$ 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
```

Chaque catégorie d'utilisateur (`owner@`, `group@`, `everyone@`) dispose d'une entrée d'ACL dans cet exemple.

Voici une description de l'ACL de ce fichier :

`0:owner@` Le propriétaire peut lire et modifier le contenu du fichier (`read_data/write_data/append_data`). Il peut également modifier les

attributs du fichier tels que les horodatages, les attributs étendus et les ACL (`write_xattr/write_attributes /write_acl`). Le propriétaire peut également modifier la propriété du fichier (`write_owner:allow`).

L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

`1:group@` Les autorisations de lecture du fichier et de ses attributs sont attribuées au groupe (`read_data/read_xattr/read_attributes/read_acl:allow`).

`2:everyone@` Les autorisations de lecture du fichier et de ses attributs sont attribués à toute personne ne correspondant ni à un utilisateur ni à un groupe (`read_data/read_xattr/read_attributes/read_acl/ synchronize:allow`). L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

Lorsqu'un répertoire est créé, en fonction de la valeur `umask`, l'ACL par défaut du répertoire est similaire à l'exemple suivant :

```
$ 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
```

Voici une description de l'ACL de ce répertoire :

`0:owner@` Le propriétaire peut lire et modifier le contenu du répertoire (`list_directory/read_data/add_file/write_data/add_subdirectory /append_data`) et lire et modifier les attributs du fichier tels que les horodatages, les attributs étendus et les ACL (`/read_xattr/write_xattr/read_attributes/write_attributes/read_acl/ write_acl`). En outre, le propriétaire peut faire des recherches dans le contenu (`execute`), supprimer un fichier ou un répertoire (`delete_child`) et modifier la possession du répertoire (`write_owner:allow`).

L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

`1:group@` Le groupe peut répertorier et lire le contenu et les attributs du répertoire. De plus, le groupe dispose d'autorisations d'exécution pour effectuer des recherches dans le contenu du répertoire (`list_directory/read_data/read_xattr/execute/read_attributes /read_acl/synchronize:allow`).

`2:everyone@` Toute personne n'étant ni un utilisateur ni un groupe dispose d'autorisations de lecture et d'exécution sur le contenu et les attributs du répertoire (`list_directory/read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow`). L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé

Vous pouvez modifier les ACL dans des fichiers ZFS à l'aide de la commande `chmod`. La syntaxe `chmod` suivante pour la modification de l'ACL utilise la *spécification acl* pour identifier le format de la liste. Pour une description de la *spécification ACL*, reportez-vous à la section "Descriptions de syntaxe pour la configuration des ACL" à la page 243.

- Ajout d'entrées d'ACL
  - Ajout d'une entrée d'ACL pour un utilisateur

    ```
    % chmod A+acl-specification filename
    ```
  - Ajout d'une entrée d'ACL par *index-ID*

    ```
    % chmod Aindex-ID+acl-specification filename
    ```

    Cette syntaxe insère la nouvelle entrée d'ACL à l'emplacement d'*index-ID* spécifié.
- Remplacement d'une entrée d'ACL

  ```
  % chmod A=acl-specification filename
  ```

  ```
  % chmod Aindex-ID=acl-specification filename
  ```
- Suppression d'entrées d'ACL
  - Suppression d'une entrée d'ACL par l'*index-ID*

    ```
    % chmod Aindex-ID- filename
    ```
  - Suppression d'une entrée d'ACL par utilisateur

    ```
    % chmod A-acl-specification filename
    ```
  - Suppression de la totalité des ACE non triviales d'un fichier

    ```
    % chmod A- filename
    ```

Les informations détaillées de l'ACL s'affichent à l'aide de la commande `ls -v`. Par exemple :

```
# 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
```

Pour obtenir des informations sur l'utilisation du format d'ACL compact, consultez “Configuration et affichage d'ACL dans des fichiers ZFS en format compact” à la page 261.

EXEMPLE 7–1 Modification des ACL triviales dans des fichiers ZFS

Cette section offre des exemples de configuration et d'affichage d'ACL triviale, ce qui signifie que seules les entrées UNIX traditionnelles, user, group et other sont incluses dans l'ACL.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `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
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

Dans l'exemple suivants, les autorisations `write_data` sont accordées au groupe `group@`.

```
# 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
```

Dans l'exemple suivant, les autorisations du fichier `file.1` sont reconfigurées sur 644.

```
# 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
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

EXEMPLE 7–2 Configuration d'ACL non triviales dans des fichiers ZFS

Cette section fournit des exemples de configuration et d'affichage d'ACL non triviales.

Dans l'exemple suivant, les autorisations `read_data/execute` sont ajoutées à l'utilisateur `gozer` dans le répertoire `test.dir`.

EXEMPLE 7–2 Configuration d'ACL non triviales dans des fichiers ZFS *(Suite)*

```
# 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
```

Dans l'exemple suivant, les autorisations `read_data/execute` sont retirées à l'utilisateur `gozer`.

```
# 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
```

EXEMPLE 7–3 Interactions entre les ACL et les autorisations dans les fichiers ZFS

Les exemples d'ACL suivants illustrent les interactions entre la configuration des ACL et la modification successive des bits d'autorisation du répertoire ou du fichier.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.2`:

```
# 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
         /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
```

Dans l'exemple suivant, les autorisations d'ACL `allow` sont supprimées de `everyone@`.

```
# 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
```

EXEMPLE 7–3 Interactions entre les ACL et les autorisations dans les fichiers ZFS *(Suite)*

Dans cette sortie, les bits d'autorisation du fichier sont réinitialisés de 644 à 640. Les autorisations de lecture de `everyone@` ont été supprimées des bits d'autorisation du fichier lorsque les autorisations "allow" des ACL ont été supprimées de `everyone@`.

Dans l'exemple suivant, l'ACL existante est remplacée par des autorisations `read_data/write_data` pour `everyone@`.

```
# 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
```

Dans cette sortie, la syntaxe `chmod` remplace effectivement l'ACL existante par les autorisations `read_data/write_data:allow` pour les autorisations de lecture/écriture pour le propriétaire, le groupe et `everyone@`. Dans ce modèle, `everyone@` spécifie l'accès à tout utilisateur ou groupe. Dans la mesure où aucune entrée d'ACL `owner@` ou `group@` n'existe pour ignorer les autorisations pour l'utilisateur ou le groupe, les bits d'autorisation sont définis sur 666.

Dans l'exemple suivant, l'ACL existante est remplacée par des autorisations de lecture pour l'utilisateur `gozer`.

```
# 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
```

Dans cette sortie, les autorisations de fichier sont calculées pour être 000 car aucune entrée d'ACL n'existe pour `owner@`, `group@`, ou `everyone@`, qui représentent les composant d'autorisation classiques d'un fichier. Le propriétaire du fichier peut résoudre ce problème en réinitialisant les autorisations (et l'ACL) comme suit :

```
# 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
```

EXEMPLE 7–4 Restauration des ACL triviales dans des fichiers ZFS

Vous pouvez utiliser la commande `chmod` pour supprimer toutes les ACL non triviales d'un fichier ou d'un répertoire.

EXEMPLE 7–4 Restauration des ACL triviales dans des fichiers ZFS *(Suite)*

Dans l'exemple suivant, deux ACE non triviales existent dans `test5.dir`.

```
# 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
```

Dans l'exemple suivant, les ACL non triviales pour les utilisateurs `gozer` et `lp` sont supprimées. L'ACL restante contient les valeurs par défaut de `owner@`, `group@` et `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
```

Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé

Vous pouvez déterminer comment les ACL sont héritées ou non dans les fichiers et répertoires. Par défaut, les ACL ne sont pas propagées. Si vous configurez une ACL non triviale dans un répertoire, aucun répertoire subséquent n'en hérite. Vous devez spécifier l'héritage d'une ACL dans un fichier ou un répertoire.

La propriété `aclinherit` peut être définie de manière globale pour un système de fichiers. Par défaut, `aclinherit` est défini sur `restricted`.

Pour plus d'informations, reportez-vous à la section “Héritage d'ACL” à la page 246.

EXEMPLE 7–5 Attribution d'héritage d'ACL par défaut

Par défaut, les ACL ne sont pas propagées par le biais d'une structure de répertoire.

EXEMPLE 7–5 Attribution d'héritage d'ACL par défaut *(Suite)*

Dans l'exemple suivant, une ACE non triviale de `read_data/write_data/execute` est appliquée pour l'utilisateur `gozer` dans le fichier `test.dir`.

```
# 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
```

Si un sous-répertoire `test.dir` est créé, l'ACE pour l'utilisateur `gozer` n'est pas propagée. L'utilisateur `gozer` n'aurait accès à `sub.dir` que si les autorisations de `sub.dir` lui accordaient un accès en tant que propriétaire de fichier, membre de groupe ou `everyone@`.

```
# 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
     2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

EXEMPLE 7–6 Attribution d'héritage d'ACL dans les fichiers et les répertoires

Cette série d'exemples identifie les ACE du fichier et du répertoire qui sont appliquées lorsque l'indicateur `file_inherit` est paramétré.

Dans cet exemple, les autorisations `read_data/write_data` sont ajoutées pour les fichiers dans le répertoire `test2.dir` pour l'utilisateur `gozer` afin qu'il dispose de l'accès en lecture à tout nouveau fichier :

```
# 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
```

EXEMPLE 7–6 Attribution d'héritage d'ACL dans les fichiers et les répertoires *(Suite)*

```
        /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
        /read_acl/synchronize:allow
```

Dans l'exemple suivant, les autorisations de l'utilisateur `gozer` sont appliquées au fichier `test2.dir/file.2` récemment créé. L'héritage d'ACL étant accordé (`read_data:file_inherit:allow`), l'utilisateur `gozer` peut lire le contenu de tout nouveau fichier.

```
# 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
```

Dans la mesure où la propriété `aclinherit` pour ce système de fichiers est paramétrée sur le mode par défaut, `restricted`, l'utilisateur `gozer` ne dispose pas de l'autorisation `write_data` pour le fichier `file.2` car l'autorisation de groupe du fichier ne le permet pas.

Notez que l'autorisation `inherit_only` appliquée lorsque les indicateurs `file_inherit` ou `dir_inherit` sont définis, est utilisée pour propager l'ACL dans la structure du répertoire. Ainsi, l'utilisateur `gozer` se voit uniquement accorder ou refuser l'autorisation des autorisations `everyone@`, à moins qu'il ne soit le propriétaire du fichier ou membre du groupe propriétaire du fichier. Par exemple :

```
# 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
```

La série d'exemples suivants identifie les ACL du fichier et du répertoire appliquées lorsque les indicateurs `file_inherit` et `dir_inherit` sont paramétrés.

Dans l'exemple suivant, l'utilisateur `gozer` se voit accorder les droits de lecture, d'écriture et d'exécution hérités des fichiers et répertoires récemment créés.

EXEMPLE 7–6 Attribution d'héritage d'ACL dans les fichiers et les répertoires *(Suite)*

```
# 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

# 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
```

Dans les exemples ci-dessus, les bits d'autorisation du répertoire parent pour `group@` et `everyone@` n'accordent pas les autorisations d'écriture et d'exécution. Par conséquent, l'utilisateur `gozer` se voit refuser ces autorisations. La propriété par défaut de `aclinherit` est `restricted`, ce qui signifie que les autorisations `write_data` et `execute` ne sont pas héritées.

Dans l'exemple suivant, l'utilisateur `gozer` se voit accorder les autorisations de lecture, d'écriture et d'exécution qui sont héritées pour les fichiers récemment créés, mais ne sont pas propagées vers tout contenu subséquent du répertoire.

```
# 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
```

EXEMPLE 7–6 Attribution d'héritage d'ACL dans les fichiers et les répertoires *(Suite)*

```
     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
```

Comme l'exemple suivant l'illustre, les autorisations `read_data/write_data/execute` de l'utilisateur `gozer` sont réduites en fonction des autorisations du groupe propriétaire.

```
# 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
         :allow
```

EXEMPLE 7–7 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through

Si la propriété `aclinherit` du système de fichiers `tank/cindy` est définie sur `passthrough`, l'utilisateur `gozer` hérite de l'ACL appliquée à `test4.dir` pour le nouveau fichier `file.5` de la manière suivante :

```
# 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
```

EXEMPLE 7–8 Héritage d'ACL avec mode d'héritage ACL défini sur Discard

Si la propriété `aclinherit` d'un système de fichiers est définie sur `discard`, il est alors possible de supprimer les ACL avec les bits d'autorisation lors d'un changement de répertoire. Par exemple :

```
# zfs set aclinherit=discard tank/cindy
# chmod A+user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
```

EXEMPLE 7–8 Héritage d'ACL avec mode d'héritage ACL défini sur Discard *(Suite)*

```
# 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
```

Si vous décidez ultérieurement de renforcer les bits d'autorisation d'un répertoire, l'ACL non triviale est supprimée. Par exemple :

```
# 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
         /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
```

EXEMPLE 7–9 Héritage d'ACL avec mode d'héritage de liste défini sur Noallow

Dans l'exemple suivant, deux ACL non triviales avec héritage de fichier sont définies. Une ACL accorde l'autorisation `read_data`, tandis qu'une autre refuse cette autorisation. Cet exemple illustre également comment spécifier deux ACE dans la même commande `chmod`.

```
# 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
```

Comme l'illustre l'exemple suivant, lors de la création d'un nouveau fichier, l'ACL qui accorde l'autorisation `read_data` est supprimée.

EXEMPLE 7–9 Héritage d'ACL avec mode d'héritage de liste défini sur Noallow *(Suite)*

```
# 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
```

Configuration et affichage d'ACL dans des fichiers ZFS en format compact

Vous pouvez définir et afficher les autorisations relatives aux fichiers ZFS en format compact utilisant 14 lettres uniques pour représenter les autorisations. Les lettres représentant les autorisations compactes sont répertoriées dans le Tableau 7–2 et le Tableau 7–4.

Vous pouvez afficher les listes d'ACL compactes pour les fichiers et les répertoires à l'aide de la commande `ls -V`. Par exemple :

```
# 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
```

La sortie d'ACL compacte est décrite comme suit :

`owner@` Le propriétaire peut lire et modifier le contenu du fichier (`rw=read_data/write_data`), (`p= append_data`). Le propriétaire peut également modifier les attributs du fichier tels que l'horodatage, les attributs étendus et les listes de contrôle d'accès (ACL) (`a=read_attributes` , `W=write_xattr`, `R= read_xattr`, `A=write_attributes` , `c=read_acl`, `C= write_acl`). De plus, le propriétaire peut modifier la propriété du fichier (`o=write_owner`).

L'autorisation d'accès `synchronize` (`s`) n'est pas implémentée pour le moment.

`group@` Les autorisations de lecture sur le fichier sont accordées au groupe sur le fichier (`r= read_data`) et les attributs du fichier (`a=read_attributes` , `R=read_xattr`, `c= read_acl`).

L'autorisation d'accès `synchronize` (`s`) n'est pas implémentée pour le moment.

`everyone@` Les autorisations de lecture sur le fichier et sur ses attributs sont accordés à toute personne n'étant ni un utilisateur ni un groupe (`r=read_data`, `a=append_data`, `R=read_xattr` , `c=read_acl` et `s= synchronize`).

L'autorisation d'accès `synchronize` (`s`) n'est pas implémentée pour le moment.

Le format d'ACL compact dispose des avantages suivants par rapport au format d'ACL détaillé :

- Les autorisations peuvent être spécifiées en tant qu'arguments de position pour la commande `chmod`.
- Les tirets (-), qui n'identifient aucune autorisation, peuvent être supprimés. Seules les lettres nécessaires doivent être spécifiées.
- Les indicateurs d'autorisations et d'héritage sont configurés de la même manière.

Pour obtenir des informations sur l'utilisation du format d'ACL détaillé, consultez "Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé" à la page 251.

EXEMPLE 7–10 Configuration et affichage des ACL en format compact

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.1` :

```
# 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
```

Dans cet exemple, les autorisations `read_data/execute` sont ajoutées à l'utilisateur `gozer` sur le fichier `file.1`.

```
# 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
```

Dans l'exemple suivant, l'utilisateur `gozer` se voit accorder les droits de lecture, d'écriture et d'exécution qui sont hérités des fichiers et répertoires récemment créés grâce à l'utilisation de l'ACL compacte.

```
# 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
```

EXEMPLE 7–10 Configuration et affichage des ACL en format compact *(Suite)*

Vous pouvez également couper et coller les autorisations et les indicateurs d'héritage à partir de la sortie `ls -V` en format `chmod` compact. Par exemple, pour dupliquer les autorisations et les indicateurs d'héritage sur `dir.2` pour l'utilisateur `gozer` vers l'utilisateur `cindy` sur `dir.2`, copiez et collez l'autorisation et les indicateurs d'héritage (`rwx-----------:fd----:allow`) dans votre commande `chmod`. Par exemple :

```
# 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
```

EXEMPLE 7–11 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through

Un système de fichiers dont la propriété `aclinherit` est définie sur `passthrough` hérite de toutes les entrées d'ACL pouvant être héritées, sans qu'aucune modification ne leur soit apportée. Lorsque cette propriété est définie sur `passthrough`, les fichiers sont créés avec un mode d'autorisation déterminé par les ACE pouvant être héritées. Si aucune ACE pouvant être héritée n'affecte le mode d'autorisation, ce mode est alors défini en fonction du mode demandé à partir de l'application.

Les exemples suivants respectent la syntaxe ACL compacte pour illustrer le processus d'héritage des bits d'autorisation en définissant le mode `aclinherit` sur la valeur `passthrough` .

Dans cet exemple, une ACL est définie sur `test1.dir` pour forcer l'héritage. La syntaxe crée une entrée d'ACL `owner@`, `group@` et `everyone@` pour les fichiers nouvellement créés. Les répertoires nouvellement créés héritent d'une entrée d'ACL `@owner`, `group@` et `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
```

Dans cet exemple, un fichier nouvellement créé hérite de l'ACL dont les fichiers nouvellement créés doivent hériter d'après ce qui a été spécifié.

EXEMPLE 7–11 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through *(Suite)*

```
# 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
```

Dans cet exemple, un répertoire nouvellement créé hérite à la fois des ACE contrôlant l'accès à ce répertoire et des ACE à appliquer ultérieurement aux enfants de ce répertoire.

```
# mkdir subdir.1
# ls -dV subdir.1
drwxrwx---+  2 root     root           2 Jun 23 16:13 subdir.1
            owner@:rwxpdDaARWcCos:fd----:allow
            group@:rwxp----------:fd----:allow
         everyone@:--------------:fd----:allow
```

Les entrées `fd----` permettent d'appliquer l'héritage et ne sont pas prises en compte lors du contrôle d'accès. Dans cet exemple, un fichier est créé avec une ACL insignifiante dans un autre répertoire ne contenant pas d'ACE héritées.

```
# 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
```

EXEMPLE 7–12 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through-X

Lorsque `aclinherit=passthrough-x` est activé, les fichiers sont créés avec l'autorisation d'exécution (`x`) pour `propriétaire@`, `groupe@` ou `tous les utilisateurs@`, mais seulement si l'autorisation d'exécution est définie dans le mode de création de fichier et dans une ACE héritable qui affecte le mode.

L'exemple suivant montre comment hériter l'autorisation d'exécution en définissant le mode `aclinherit` sur `passthrough-x`.

```
# zfs set aclinherit=passthrough-x tank/cindy
```

L'ACL suivante est définie sur `/tank/cindy/test1.dir` pour permettre l'héritage des ACL exécutables pour les fichiers de `owner@`.

```
# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow test1.dir
# ls -Vd test1.dir
```

EXEMPLE 7–12 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through-X *(Suite)*

```
drwxrwx---+  2 root     root           2 Jun 23 16:17 test1.dir
           owner@:rwxpdDaARWcCos:fd----:allow
           group@:rwxp----------:fd----:allow
        everyone@:--------------:fd----:allow
```

Un fichier (`file1`) est créé avec les autorisations demandées `0666`. Les autorisations obtenues sont `0660`. L'autorisation d'exécution n'était pas héritée car le mode de création ne le requérait pas.

```
# 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
```

Ensuite, un fichier exécutable appelé `t` est généré à l'aide du compilateur `cc` dans le répertoire `testdir`.

```
# 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
```

Les autorisations obtenues sont `0770` car `cc` a demandé des autorisations `0777`, ce qui a entraîné l'héritage de l'autorisation d'exécution à partir des entrées `propriétaire@`, `groupe@` et `tous les utilisateurs@`.

EXEMPLE 7–13 Interactions entre les ACL et les opérations `chmod` sur les fichiers ZFS

Les exemples suivants illustrent l'incidence de certaines valeurs des propriétés `aclmode` et `aclinherit` sur l'interaction des ACL existantes avec une opération `chmod` modifiant les autorisations de répertoire ou de fichier en vue de restreindre ou d'augmenter les autorisations d'ACL existantes à des fins de conformité avec le groupe propriétaire.

Dans cet exemple, la propriété `aclmode` est définie sur `mask` et la propriété `aclinherit` sur `restricted`. Les autorisations d'ACL sont affichées en mode compact dans cet exemple, ce qui permet de mieux repérer les modifications apportées aux autorisations.

Paramètres de propriété du fichier et des groupes et autorisations d'ACL initiaux :

```
# 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
```

EXEMPLE 7–13 Interactions entre les ACL et les opérations chmod sur les fichiers ZFS *(Suite)*

```
             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
```

Une opération chown modifie la propriété du fichier `file.1` et la sortie est visible par l'utilisateur propriétaire, amy. Par exemple :

```
# 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
```

Les opérations chmod suivantes font passer les autorisations à un mode plus restrictif. Dans cet exemple, les autorisations d'ACL modifiées du groupe `sysadmin` et du groupe `staff` n'excèdent pas les autorisations du groupe propriétaire.

```
$ 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
```

L'opération chmod suivante fait passer les autorisations à un mode moins restrictif. Dans cet exemple, les autorisations d'ACL modifiées du groupe `sysadmin` et du groupe `staff` sont restaurées pour accorder les mêmes autorisations que celles du groupe propriétaire.

```
$ 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
```

◆ ◆ ◆ CHAPITRE 8

Administration déléguée de ZFS dans Oracle Solaris

Ce chapitre explique comment utiliser les fonctions d'administration déléguée pour permettre aux utilisateurs ne disposant pas des autorisations nécessaires d'effectuer des tâches d'administration de ZFS.

Ce chapitre contient les sections suivantes :

- "Présentation de l'administration déléguée de ZFS" à la page 267
- "Délégation d'autorisations ZFS" à la page 268
- "Affichage des autorisations ZFS déléguées (exemples)" à la page 276
- "Délégation d'autorisations ZFS (exemples)" à la page 272
- "Suppression des autorisations ZFS déléguées (exemples)" à la page 278

Présentation de l'administration déléguée de ZFS

L'administration déléguée de ZFS vous permet de distribuer des autorisations précises à des utilisateurs ou à des groupes spécifiques, voire à tous les utilisateurs. Deux types d'autorisations déléguées sont prises en charge :

- Les autorisations individuelles suivantes peuvent être explicitement déléguées : autorisation de création (`create`), autorisation de destruction (`destroy`), autorisation de montage (`mount`), autorisation de créer des instantanés (`snapshot`), etc.
- Des groupes d'autorisations appelés *jeux d'autorisations* peuvent être définis. Si un jeu d'autorisations est modifié, tout utilisateur de ce jeu de d'autorisations est automatiquement affecté par ces modifications. Les jeux d'autorisations commencent par le symbole `@` et sont limités à 64 caractères. Les caractères suivant le symbole `@` dans le nom de jeu ont les mêmes restrictions que ceux des noms de systèmes de fichiers ZFS standard.

L'administration déléguée de ZFS offre des fonctions similaires au modèle de sécurité RBAC. La délégation ZFS offre les avantages suivants pour la gestion des pools de stockage et systèmes de fichiers ZFS :

- Les autorisations sont transférées avec le pool de stockage ZFS lorsqu'un pool est migré.

- L'héritage dynamique vous permet de contrôler la propagation des autorisations dans les systèmes de fichiers.
- Il est possible de définir une configuration de manière à ce que seul le créateur d'un système de fichiers puisse détruire celui-ci.
- Les autorisations peuvent être déléguées à des systèmes de fichiers spécifiques. Tout nouveau système de fichiers peut automatiquement récupérer des autorisations.
- Simplifie l'administration de systèmes de fichiers en réseau (NFS, Network File System). Un utilisateur disposant d'autorisations explicites peut par exemple créer un instantané sur un système NFS dans le répertoire `.zfs/snapshot` approprié.

Considérez l'utilisation de l'administration déléguée pour la répartition des tâches ZFS. Pour plus d'informations sur l'utilisation de RBAC pour gérer les tâches d'administration générales dans Oracle Solaris, reportez-vous à Partie III, “Roles, Rights Profiles, and Privileges” du manuel *System Administration Guide: Security Services*.

Désactivation des droits délégués de ZFS

La propriété `delegation` du pool permet de contrôler les fonctions d'administration déléguées. Par exemple :

```
# 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
```

Par défaut, la propriété `delegation` est activée.

Délégation d'autorisations ZFS

Vous pouvez utiliser la commande `zfs allow` pour déléguer les autorisations sur les systèmes de fichiers ZFS vers des utilisateurs non-root en utilisant l'une des méthodes suivantes :

- Vous pouvez déléguer des autorisations individuelles à un utilisateur, à un groupe, voire à tous les utilisateurs.
- Vous pouvez déléguer des groupes d'autorisations individuelles sous forme de *jeu d'autorisations* à un utilisateur, à un groupe, voire à tous les utilisateurs.
- Vous pouvez déléguer des autorisations localement, soit uniquement au système de fichiers actuel, soit à l'ensemble de ses descendants.

Le tableau suivant décrit les opérations pouvant être déléguées et toute autorisation dépendante requise pour réaliser ces opérations déléguées.

Autorisation (sous-commande)	Description	Dépendances
`allow`	Autorisation d'accorder des autorisations qui vous ont été octroyées à un autre utilisateur.	Nécessité de disposer de l'autorisation en cours d'octroi.
`clone`	Autorisation de cloner tout instantané du jeu de données.	Nécessité de disposer des autorisations `create` et `mount` sur le système de fichiers d'origine.
`create`	Autorisation de créer des jeux de données descendants.	Nécessité de disposer de l'autorisation `mount`.
`destroy`	Autorisation de détruire un jeu de données.	Nécessité de disposer de l'autorisation `mount`.
`diff`	Autorisation d'identifier les chemins d'accès à l'intérieur d'un jeu de données.	Les utilisateurs non-root ont besoin de cette autorisation pour utiliser la commande `zfs diff`.
`hold`	Autorisation de conservation d'un instantané.	
`mount`	Autorisation de monter et démonter un système de fichiers, et de créer et détruire les liens vers des périphériques de volume.	
`promote`	Autorisation de promouvoir le clonage d'un jeu de données.	Nécessité de disposer également des autorisations `mount` et `promote` sur le système de fichiers d'origine.
`receive`	Autorisation de créer des systèmes de fichiers descendants à l'aide de la commande `zfs receive`.	Nécessité de disposer également des autorisations `mount` et `create`.
`release`	Autorisation de libérer un instantané conservé, ce qui peut détruire l'instantané.	
`rename`	Autorisation de renommer un jeu de données.	Nécessité de disposer également des autorisations `create` et `mount` sur le nouveau parent.
`restauration`	Autorisation de restaurer un instantané.	
`send`	Autorisation d'envoyer un flux d'instantané.	

Autorisation (sous-commande)	Description	Dépendances
`share`	Autorisation de partager et de départager un système de fichiers	`share` et `sharenfs` sont tous les deux nécessaires pour créer un partage NFS. `share` et `sharesmb` sont tous les deux nécessaires pour créer un partage SMB.
`snapshot`	Autorisation de créer un instantané d'un jeu de données.	

Vous pouvez déléguer le jeu d'autorisations suivant mais une autorisation peut être limitée à l'accès, à la lecture ou à la modification :

- `groupquota`
- `groupused`
- `userprop`
- `userquota`
- `userused`

Vous pouvez en outre déléguer l'administration des propriétés ZFS suivantes à des utilisateurs non-root :

- `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

Certaines de ces propriétés ne peuvent être définies qu'à la création d'un jeu de données. Pour une description de ces propriétés, reportez-vous à la section "Présentation des propriétés ZFS" à la page 183.

Délégation des autorisations ZFS (zfs allow)

La syntaxe de zfs allow est la suivante :

zfs allow -[ldugecs] everyone|*user*|*group*[,...] *perm*|@*setname*,...] *filesystem*| *volume*

La syntaxe de zfs allow suivante (en gras) identifie les utilisateurs auxquels les autorisations sont déléguées :

zfs allow [-uge]|user|group|everyone [,...] *filesystem* | *volume*

Vous pouvez spécifier plusieurs entrées sous forme de liste séparée par des virgules. Si aucune option -uge n'est spécifiée, l'argument est interprété en premier comme le mot-clé everyone, puis comme un nom d'utilisateur et enfin, comme un nom de groupe. Pour spécifier un utilisateur ou un groupe nommé "everyone", utilisez l'option -u ou l'option -g. Pour spécifier un groupe portant le même nom qu'un utilisateur, utilisez l'option -g. L'option -c délègue des autorisations create-time.

La syntaxe de zfs allow suivante (en gras) identifie la méthode de spécification des autorisations et jeux d'autorisations :

zfs allow [-s] ... perm|@setname [,...] *filesystem* | *volume*

Vous pouvez spécifier plusieurs autorisations sous forme de liste séparée par des virgules. Les noms d'autorisations sont identiques aux sous-commandes et propriétés ZFS. Pour plus d'informations, reportez-vous à la section précédente.

Les autorisations peuvent être regroupées en *jeux d'autorisations* et sont identifiées par l'option -s. Les jeux d'autorisations peuvent être utilisés par d'autres commandes zfs allow pour le système de fichiers spécifié et ses descendants. Les jeux d'autorisations sont évalués

dynamiquement et de ce fait, toute modification apportée à un jeu est immédiatement mise à jour. Les jeux d'autorisations doivent se conformer aux mêmes critères d'attribution de noms que les systèmes de fichiers ZFS, à ceci près que leurs noms doivent commencer par le caractère arobase (@) et ne pas dépasser 64 caractères.

La syntaxe de `zfs allow` suivante (en gras) identifie la méthode de délégation des autorisations :

`zfs allow [-ld]` `...` `...` *filesystem* | *volume*

L'option `-l` indique que les autorisations sont accordées au système de fichiers spécifié mais pas à ses descendants, à moins de spécifier également l'option `-d`. L'option `-d` indique que les autorisations sont accordées aux systèmes de fichiers descendants mais pas à ce système de fichiers, à moins de spécifier également l'option `-l`. Si aucune option n'est spécifiée, les autorisations sont accordées au système de fichiers ou au volume ainsi qu'à ses descendants.

Suppression des autorisations déléguées de ZFS (`zfs unallow`)

Vous pouvez supprimer des autorisations précédemment déléguées à l'aide de la commande `zfs unallow`.

Supposons par exemple que vous déléguiez les autorisations `create`, `destroy`, `mount` et `snapshot` de la manière suivante :

```
# 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
```

Pour supprimer ces autorisations, vous devez respecter la syntaxe suivante :

```
# zfs unallow cindy tank/home/cindy
# zfs allow tank/home/cindy
```

Délégation d'autorisations ZFS (exemples)

EXEMPLE 8–1 Délégation d'autorisations à un utilisateur individuel

Lorsque vous déléguez les autorisations `create` et `mount` à un utilisateur individuel, vous devez vous assurer que cet utilisateur dispose d'autorisations sur le point de montage sous-jacent.

Par exemple, pour déléguer à l'utilisateur `mark` les autorisations `create` et `mount` sur le système de fichiers `tank`, définissez au préalable ces autorisations :

EXEMPLE 8–1 Délégation d'autorisations à un utilisateur individuel *(Suite)*

```
# chmod A+user:mark:add_subdirectory:fd:allow /tank/home
```

Utilisez ensuite la commande `zfs allow` pour déléguer les autorisations `create`, `destroy` et `mount`. Par exemple :

```
# zfs allow mark create,destroy,mount tank/home
```

L'utilisateur `mark` peut dorénavant créer ses propres systèmes de fichiers dans le système de fichiers `tank/home`. Par exemple :

```
# su mark
mark$ zfs create tank/home/mark
mark$ ^D
# su lp
$ zfs create tank/home/lp
cannot create 'tank/home/lp': permission denied
```

EXEMPLE 8–2 Délégation des autorisations de création (create) et de destruction (destroy) à un groupe

L'exemple suivant décrit comment configurer un système de fichiers de manière à ce que tout membre du groupe `staff` puisse créer et monter des systèmes de fichiers dans le système de fichiers `tank/home`, et détruire ses propres systèmes de fichiers. Toutefois, les membres du groupe `staff` ne sont pas autorisés à détruire les systèmes de fichiers des autres utilisateurs.

```
# 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
```

EXEMPLE 8–3 Délégation d'autorisations au niveau approprié d'un système de fichiers

Assurez-vous de déléguer les autorisations aux utilisateurs au niveau approprié du système de fichiers. Par exemple, les autorisations `create`, `destroy` et `mount` pour les systèmes de fichiers locaux et descendants sont déléguées à l'utilisateur `mark`. L'autorisation locale de créer un instantané du système de fichiers `tank/home` a été délégué à l'utilisateur `mark`, mais pas celle de créer un instantané de son propre système de fichiers. L'autorisation `snapshot` ne lui a donc pas été déléguée au niveau approprié du système de fichiers.

EXEMPLE 8–3 Délégation d'autorisations au niveau approprié d'un système de fichiers *(Suite)*

```
# 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
```

Pour déléguer à l'utilisateur `mark` cette autorisation au niveau du système de fichiers descendants, utilisez l'option `zfs allow -d`. Par exemple :

```
# 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
```

L'utilisateur `mark` ne peut maintenant créer un instantané qu'à un niveau inférieur du système de fichiers `tank/home`.

EXEMPLE 8–4 Définition et utilisation d'autorisations déléguées complexes

Vous pouvez déléguer des autorisations spécifiques à des utilisateurs ou des groupes. Par exemple, la commande `zfs allow` suivante délègue des autorisations spécifiques au groupe `staff`. En outre, les autorisations `destroy` et `snapshot` sont déléguées après la création de systèmes de fichiers `tank/home`.

```
# 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
```

EXEMPLE 8–4 Définition et utilisation d'autorisations déléguées complexes *(Suite)*

Etant donné que l'utilisateur `mark` est membre du groupe `staff`, il peut créer des systèmes de fichiers dans `tank/home`. En outre, l'utilisateur `mark` peut créer un instantané de `tank/home/mark2` parce qu'il dispose des autorisations spécifiques pour le faire. Par exemple :

```
# 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
```

L'utilisateur `mark` ne peut pas créer d'instantané dans `tank/home/mark` parce qu'il ne dispose pas des autorisations spécifiques pour le faire. Par exemple :

```
$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

Dans cet exemple, l'utilisateur `mark` possède l'autorisation `create` dans son répertoire personnel, ce qui signifie qu'il peut créer des instantanés. Ce scénario s'avère utile lorsque votre système de fichiers est monté sur un système NFS.

```
$ 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
```

EXEMPLE 8–5 Définition et utilisation d'un jeu d'autorisations délégué ZFS

L'exemple suivant décrit comment créer un jeu d'autorisations intitulé `@myset` et délègue ce jeu d'autorisations ainsi que l'autorisation de `renommage` au groupe `staff` pour le système de fichiers `tank`. L'utilisateur `cindy`, membre du groupe `staff`, a l'autorisation de créer un système de fichiers dans `tank`. Par contre, l'utilisateur `lp` ne dispose pas de cette autorisation de création de systèmes de fichiers dans `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
```

Affichage des autorisations ZFS déléguées (exemples)

Vous pouvez vous servir de la commande suivante pour afficher les autorisations :

```
# zfs allow dataset
```

Cette commande affiche les autorisations définies ou accordées au jeu de données spécifié. La sortie contient les composants suivants :

- Jeux d'autorisations
- Autorisations individuelles ou autorisations à la création
- Jeu de données local
- Jeux de données locaux et descendants
- Jeux de données descendants uniquement

EXEMPLE 8–6 Affichage des autorisations d'administration déléguées de base

La sortie suivante indique que l'utilisateur `cindy` dispose des autorisations `create`, `destroy`, `mount` et snapshot sur le système de fichiers `tank/cindy`.

```
# zfs allow tank/cindy
       -------------------------------------------------------------
       Local+Descendent permissions on (tank/cindy)
               user cindy create,destroy,mount,snapshot
```

EXEMPLE 8–7 Affichage des autorisations d'administration déléguée complexes

La sortie de cet exemple indique les autorisations suivantes sur les systèmes de fichiers `pool/fred` et `pool`.

Pour le système de fichiers `pool/fred` :

- Deux jeux d'autorisations sont définis :
  - `@eng` (`create`, `destroy` , `snapshot`, `mount`, `clone` , `promote`, `rename`)
  - `@simple` (`create`, `mount`)
- Les autorisations à la création sont définies pour le jeu d'autorisations `@eng` et la propriété `mountpoint`. "A la création" signifie qu'une fois qu'un jeu de systèmes de fichiers est créé, le jeu d'autorisations `@eng` et l'autorisation de définir la propriété `mountpoint` sont déléguées.
- Le jeu d'autorisations `@eng` est délégué à l'utilisateur `tom` et les autorisations `create`, `destroy` et `mount` pour les systèmes de fichiers locaux sont déléguées à l'utilisateur `joe`.
- Le jeu d'autorisations `@basic` ainsi que les autorisations `share` et `rename` pour les systèmes de fichiers locaux et descendants sont délégués à l'utilisateur `fred`.
- Le jeu d'autorisations `@basic` pour les systèmes de fichiers descendants uniquement est délégué à l'utilisateur `barney` et au groupe `staff`.

Pour le système de fichiers `pool` :

- Le jeu d'autorisations `@simple` (create, destroy, mount) est défini.
- Le jeu d'autorisations sur le système de fichiers local `@simple` est accordé au groupe `staff`.

La sortie de cet exemple est la suivante :

```
$ 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:
        user fred @basic,share,rename
```

EXEMPLE 8–7 Affichage des autorisations d'administration déléguée complexes *(Suite)*

```
        user barney @basic
        group staff @basic
---- Permissions on pool --------------------------------------------
Permission sets:
        @simple create,destroy,mount
Local permissions:
        group staff @simple
```

Suppression des autorisations ZFS déléguées (exemples)

Vous pouvez utiliser la commande `zfs unallow` pour supprimer des autorisations déléguées. Par exemple, l'utilisateur `cindy` possède les autorisations `create`, `destroy`, `mount` et `snapshot` sur le système de fichiers `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
```

La syntaxe suivante de la commande `zfs unallow` supprime l'autorisation de réaliser des instantanés (`snapshot`) du système de fichiers `tank/home/cindy` accordée à l'utilisateur `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
```

Autre exemple : l'utilisateur `mark` possède les autorisations suivantes sur le système de fichiers `tank/home/mark` :

```
# zfs allow tank/home/mark
---- Permissions on tank/home/mark ----------------------------------
Local+Descendent permissions:
        user mark create,destroy,mount
-------------------------------------------------------------
```

La syntaxe suivante de la commande `zfs unallow` supprime toutes les autorisations accordées à l'utilisateur `mark` pour le système de fichiers `tank/home/mark` :

```
# zfs unallow mark tank/home/mark
```

La syntaxe suivante de la commande `zfs unallow` supprime un jeu d'autorisations sur le système de fichiers `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
```

◆ ◆ ◆ CHAPITRE 9

9 Rubriques avancées Oracle Solaris ZFS

Ce chapitre décrit les volumes ZFS, l'utilisation de ZFS dans un système Solaris avec zones installées, les pools root de remplacement ZFS et les profils de droits ZFS.

Ce chapitre contient les sections suivantes :

- "Volumes ZFS" à la page 281
- "Utilisation de ZFS dans un système Solaris avec zones installées" à la page 284
- "Utilisation de pools root ZFS de remplacement" à la page 290

Volumes ZFS

Un volume ZFS est un jeu de données qui représente un périphérique en mode bloc. Les volumes ZFS sont identifiés en tant que périphériques dans le répertoire `/dev/zvol/{dsk,rdsk}/pool`.

Dans l'exemple suivant, un volume ZFS de 5 GO portant le nom `tank/vol` est créé :

```
# zfs create -V 5gb tank/vol
```

Lors de la création d'un volume, une réservation est automatiquement définie sur la taille initiale du volume pour éviter tout comportement inattendu. Si, par exemple, la taille du volume diminue, les données risquent d'être altérées. Vous devez faire preuve de prudence lors de la modification de la taille du volume.

En outre, si vous créez un instantané pour un volume modifiant la taille de ce dernier, cela peut provoquer des incohérences lorsque vous tentez d'annuler (roll back) l'instantané ou de créer un clone à partir de l'instantané.

Pour de plus amples informations concernant les propriétés de systèmes de fichiers applicables aux volumes, reportez-vous au Tableau 5–1.

Pour afficher les informations de propriétés de volumes ZFS à l'aide de la commande `zfs get` ou `zfs get all`. Par exemple :

```
# zfs get all tank/vol
```

Si un point d'interrogation (?) s'affiche dans `volsize` de la sortie `zfs get`, cela signifie qu'une valeur est inconnue car une erreur d'E/S s'est produite. Par exemple :

```
# zfs get -H volsize tank/vol
tank/vol        volsize ?       local
```

Une erreur d'E/S indique généralement un problème lié à un périphérique de pool. Pour plus d'informations sur la résolution des problèmes liés aux périphériques, consultez "Identification des problèmes avec les pools de stockage ZFS" à la page 296.

En cas d'utilisation d'un système Solaris avec zones installées, la création ou le clonage d'un volume ZFS dans une zone non globale est impossible. Si vous tentez d'effectuer cette action, cette dernière échouera. Pour obtenir des informations relatives à l'utilisation de volumes ZFS dans une zone globale, reportez-vous à la section "Ajout de volumes ZFS à une zone non globale" à la page 287.

Utilisation d'un volume ZFS en tant que périphérique de swap ou de vidage

Lors de l'installation d'un système de fichiers root ZFS ou d'une migration à partir d'un système de fichiers root UFS, un périphérique de swap est créé sur un volume ZFS du pool root ZFS. Par exemple :

```
# swap -l
swapfile                  dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap 253,3        16  8257520  8257520
```

Lors de l'installation d'un système de fichiers root ZFS ou d'une migration à partir d'un système de fichiers root UFS, un périphérique de vidage est créé sur un volume ZFS du pool root ZFS. Le périphérique de vidage ne nécessite aucune administration une fois configuré. Par exemple :

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/
  Savecore enabled: yes
```

Si vous devez modifier votre zone de swap ou votre périphérique de vidage après l'installation du système, utilisez les commandes `swap` et `dumpadm` de la même manière que dans les versions précédentes de Solaris. Si vous tentez de créer un autre volume de swap, créez un volume ZFS d'une taille spécifique et activez le swap sur le périphérique. Ajoutez ensuite une entrée pour le nouveau périphérique de swap dans le fichier `/etc/vfstab`. Par exemple :

```
# 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
```

N'effectuez pas de swap vers un fichier dans un système de fichiers ZFS. La configuration de fichier swap ZFS n'est pas prise en charge.

Pour plus d'informations sur l'ajustement de la taille des volumes de swap et de vidage, reportez-vous à la section "Ajustement de la taille de vos périphériques de swap et de vidage ZFS" à la page 159.

Utilisation d'un volume ZFS en tant que cible iSCSI Solaris

Vous pouvez facilement créer un volume ZFS en tant que cible iSCSI en configurant la propriété `shareiscsi` sur le volume. Par exemple :

```
# 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
```

Une fois la cible iSCSI créée, configurez l'initiateur iSCSI. Pour de plus amples informations sur les cibles iSCSI et les initiateurs Solaris, reportez-vous au Chapitre 12, "Configuring Oracle Solaris iSCSI Targets (Tasks)" du manuel *System Administration Guide: Devices and File Systems*.

Remarque – La commande `iscsitadm` permet la création et la gestion de cibles Solaris iSCSI. Si vous avez configuré la propriété `shareiscsi` dans un volume ZFS, n'utilisez pas la commande `iscsitadm` pour créer le même périphérique cible. Vous pouvez également dupliquer les informations des cibles sur ce même périphérique.

Vous pouvez gérer un volume ZFS configuré en tant que cible iSCSI de la même façon qu'un autre jeu de données ZFS. Cependant, les opérations de `renommage`, d'`exportation` et d'`importation` fonctionnent de façon différente pour les cibles iSCSI.

- Lors du renommage d'un volume ZFS, le nom de la cible iSCSI ne change pas. Par exemple :

  ```
  # 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
  ```

- L'exportation d'un pool contenant un volume ZFS entraîne la suppression de la cible. L'importation d'un pool contenant un volume ZFS entraîne le partage de la cible. Par exemple :

```
# 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
```

L'ensemble des informations de configuration de cible iSCSI est stocké dans le jeu de données. Tout comme un système de fichiers NFS partagé, une cible iSCSI importée dans un système différent est partagée adéquatement.

Utilisation de ZFS dans un système Solaris avec zones installées

Les sections suivantes décrivent l'utilisation d'un système de fichiers ZFS sur un système avec des zones Oracle Solaris :

- "Ajout de systèmes de fichiers ZFS à une zone non globale" à la page 285
- "Délégation de jeux de données à une zone non globale" à la page 286
- "Ajout de volumes ZFS à une zone non globale" à la page 287
- "Utilisation de pools de stockage ZFS au sein d'une zone" à la page 288
- "Gestion de propriétés ZFS au sein d'une zone" à la page 288
- "Explication de la propriété zoned" à la page 289

Pour plus d'informations sur la configuration de zones d'un système de fichiers root ZFS qui va être migré ou auquel des patchs vont être appliqués à l'aide de Solaris Live Upgrade, reportez-vous à la section "Utilisation de Live Upgrade pour migrer ou mettre à niveau un système avec zones (Solaris 10 10/08)" à la page 142 ou "Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 10 5/09 ou supérieure)" à la page 147.

Tenez compte des points suivants lors de l'association de jeux de données à des zones :

- Il est possible d'ajouter un système de fichiers ou un clone ZFS à une zone non globale en déléguant ou non le contrôle administratif.
- Vous pouvez ajouter un volume ZFS en tant que périphérique à des zones non globales.
- L'association d'instantanés ZFS à des zones est impossible à l'heure actuelle.

Dans les sections suivantes, le terme jeu de données ZFS fait référence à un système de fichiers ou à un clone.

L'ajout d'un jeu de données permet à la zone non globale de partager l'espace avec la zone globale, mais l'administrateur de zone ne peut pas contrôler les propriétés ou créer de nouveaux systèmes de fichiers dans la hiérarchie de systèmes de fichiers sous-jacents. Cette opération est identique à l'ajout de tout autre type de système de fichiers à une zone. Effectuez-la lorsque vous souhaitez simplement partager de l'espace commun.

ZFS autorise également la délégation de jeux de données à une zone non globale, ce qui permet à l'administrateur de zone de contrôler parfaitement le jeu de données et ses enfants. L'administrateur de zone peut créer et détruire les systèmes de fichiers ou les clones au sein de ce jeu de données et modifier les propriétés des jeux de données. L'administrateur de zone ne peut pas affecter des jeux de données qui n'ont pas été ajoutés à la zone, y compris ceux qui dépassent les quotas de niveau supérieur du jeu de données délégué.

Tenez compte des points suivants lorsque vous utilisez ZFS sur un système sur lequel des zones Oracle Solaris sont installées :

- La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`.
- N'ajoutez pas de jeu de données ZFS à une zone non globale lorsque celle-ci est configurée. Ajoutez plutôt un jeu de données ZFS une fois la zone installée.
- Lorsqu'un emplacement source `zonepath` et l'emplacement cible `zonepath` résident tous deux dans un système de fichiers ZFS et se trouvent dans le même pool, la commande `zoneadm clone` utilise dorénavant automatiquement le clone ZFS pour cloner une zone. La commande `zoneadm clone` crée un instantané ZFS de la source de l'emplacement `zonepath` et configure l'emplacement `zonepath` cible. Vous ne pouvez pas utiliser la commande `zfs clone` pour cloner une zone. Pour plus d'informations, reportez-vous à la Partie II, “Zones” du manuel *Guide d'administration système : Conteneurs Oracle Solaris-Gestion des ressources et Oracle Solaris Zones*.
- Si vous déléguez un système de fichiers ZFS à une zone non globale, vous devez supprimer ce système de fichiers de la zone non globale avant d'utiliser Oracle Solaris Live Upgrade. Sinon, l'opération Oracle Solaris Live Upgrade échoue en raison d'une erreur de système de fichiers en lecture seule.

Ajout de systèmes de fichiers ZFS à une zone non globale

Vous pouvez ajouter un système de fichiers ZFS en tant que système de fichiers générique lorsqu'il s'agit simplement de partager de l'espace avec la zone globale. La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`. Par exemple, si le système de fichiers `tank/zone/zion` doit être ajouté à une zone non globale, définissez comme suit la propriété `mountpoint` dans la zone globale :

```
# zfs set mountpoint=legacy tank/zone/zion
```

La sous-commande `add fs` de la commande `zonecfg` permet d'ajouter un système de fichiers ZFS à une zone non globale.

Dans l'exemple suivant, un système de fichiers ZFS est ajouté à une zone non globale par un administrateur global de la zone globale :

```
# 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
```

Cette syntaxe permet d'ajouter le système de fichiers ZFS `tank/zone/zion` à la zone `zion` déjà configurée et montée sur `/opt/data`. La propriété `mountpoint` du système de fichiers doit être définie sur `legacy` et le système de fichiers ne peut pas être déjà monté à un autre emplacement. L'administrateur de zone peut créer et détruire des fichiers au sein du système de fichiers. Le système de fichiers ne peut pas être remonté à un autre emplacement, tout comme l'administrateur ne peut pas modifier les propriétés suivantes du système de fichiers : `atime`, `readonly`, `compression`, etc. L'administrateur de zone globale est chargé de la configuration et du contrôle des propriétés du système de fichiers.

Pour plus d'informations sur la commande `zonecfg` et la configuration des types de ressource avec `zonecfg`, reportez-vous à la Partie II, “Zones” du manuel *Guide d'administration système : Conteneurs Oracle Solaris-Gestion des ressources et Oracle Solaris Zones*.

Délégation de jeux de données à une zone non globale

Si l'objectif principal est de déléguer l'administration du stockage d'une zone, le système de fichiers ZFS prend en charge l'ajout de jeux de données à une zone non globale à l'aide de la sous-commande `add dataset` de la commande `zonecfg`.

Dans l'exemple suivant, un système de fichiers ZFS est délégué à une zone non globale par un administrateur global dans la zone globale.

```
# zonecfg -z zion
zonecfg:zion> add dataset
zonecfg:zion:dataset> set name=tank/zone/zion
zonecfg:zion:dataset> end
```

Contrairement à l'ajout d'un système de fichiers, cette syntaxe entraîne la visibilité du système de fichiers ZFS `tank/zone/zion` dans la zone `zion` déjà configurée. L'administrateur de zone peut définir les propriétés de système de fichiers, et créer des systèmes de fichiers descendants. En outre, l'administrateur des zones peut créer des instantanés ainsi que des clones, et contrôler la totalité de la hiérarchie du système de fichiers.

Contrairement à l'ajout d'un système de fichiers, cette syntaxe entraîne la visibilité du système de fichiers ZFS `tank/zone/zion` dans la zone `zion` déjà configurée. Dans la zone `zion`, ce système de fichiers n'est pas accessible en tant que `tank/zone/zion`, mais en tant que *virtual pool* nommé `tank`. L'alias du système de fichiers délégué fournit à la zone en tant que pool virtuel une vue du pool d'origine. La propriété d'alias indique le nom du pool virtuel. Si aucun alias n'est précisé, un alias par défaut correspondant au dernier composant du nom du système de fichiers est utilisé. Si aucun alias n'avait été indiqué, l'alias par défaut aurait été `zion` dans l'exemple ci-dessus.

Dans les jeux de données délégués, l'administrateur des zones peut définir les propriétés des systèmes de fichiers et créer des systèmes de fichiers descendants. En outre, l'administrateur des zones peut créer des instantanés ainsi que des clones, et contrôler la totalité de la hiérarchie du système de fichiers. Si des volumes ZFS sont créés au sein de systèmes de fichier délégués, ils risquent d'entrer en conflit avec les volumes ZFS ajoutés en tant que ressources de périphériques. Pour de plus amples informations, consultez la section suivante.

Si vous utilisez Oracle Solaris Live Upgrade pour mettre à niveau l'environnement d'initialisation du système de fichiers ZFS avec des zones non globales, supprimez au préalable tous les jeux de données délégués. Sinon, l'opération Oracle Solaris Live Upgrade échoue en raison d'une erreur de système de fichiers en lecture seule. Par exemple :

```
zonecfg:zion>
zonecfg:zion> remove dataset name=tank/zone/zion
zonecfg:zion1> exit
```

Pour de plus amples informations relatives aux actions autorisées au sein des zones, reportez-vous à la section “Gestion de propriétés ZFS au sein d'une zone” à la page 288.

Ajout de volumes ZFS à une zone non globale

Les volumes ZFS ne peuvent pas être ajoutés à une zone non globale à l'aide de la sous-commande `add dataset` de la commande `zonecfg`. Il est cependant possible d'ajouter des volumes à une zone à l'aide de la sous-commande `add device` de la commande `zonecfg`.

Dans l'exemple suivant, un volume ZFS est ajouté à une zone non globale par un administrateur global de la zone globale :

```
# 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
```

Cette syntaxe ajoute le volume `tank/vol` à la zone `zion` .

Notez que l'ajout d'un volume brut à une zone comporte des risques de sécurité implicites, même si le volume ne correspond pas à un périphérique physique. L'administrateur risque notamment de créer des systèmes de fichiers non conformes qui généreraient des erreurs graves dans le système en cas de tentative de montage. Pour de plus amples informations sur l'ajout de périphériques à de zones et les risques de sécurités associés, reportez-vous à la section "Explication de la propriété zoned" à la page 289.

Pour plus d'informations sur l'ajout des périphériques à des zones, reportez-vous à la Partie II, "Zones" du manuel *Guide d'administration système : Conteneurs Oracle Solaris-Gestion des ressources et Oracle Solaris Zones*.

Utilisation de pools de stockage ZFS au sein d'une zone

Il est impossible de créer ou de modifier des pools de stockage ZFS au sein d'une zone. Le modèle d'administration délégué centralise le contrôle de périphériques de stockage physique au sein de la zone globale et le contrôle du stockage virtuel dans les zones non globales. Bien qu'un jeu de données au niveau du pool puisse être ajouté à une zone, toute commande modifiant les caractéristiques physiques du pool, comme la création, l'ajout ou la suppression de périphériques est interdite au sein de la zone. Même si les périphériques physiques sont ajoutés à une zone à l'aide de la sous-commande `add device` de la commande `zonecfg`, ou si les fichiers sont utilisés, la commande `zpool` n'autorise pas la création de nouveaux pools au sein de la zone.

Gestion de propriétés ZFS au sein d'une zone

Après avoir délégué un jeu de données à une zone, l'administrateur de zone peut contrôler les propriétés spécifiques au jeu de données. Lorsqu'un jeu de données est délégué à une zone, tous les ancêtres s'affichent en tant que jeux de données en lecture seule, alors que le jeu de données lui-même, ainsi que tous ses descendants, est accessible en écriture. Considérez par exemple la configuration suivante :

```
global# zfs list -Ho name
tank
tank/home
tank/data
tank/data/matrix
tank/data/zion
tank/data/zion/home
```

En cas d'ajout de `tank/data/zion` à une zone, chaque jeu de données dispose des propriétés suivantes :

Jeu de données	Visible	Accessible en écriture	Propriétés immuables
`tank`	Oui	Non	-
`tank/home`	Non	-	-
`tank/data`	Oui	Non	-
`tank/data/matrix`	Non	-	-
`tank/data/zion`	Oui	Oui	`sharenfs`, `zoned`, `quota`, `reservation`
`tank/data/zion/home`	Oui	Oui	`sharenfs`, `zoned`

Notez que chaque parent de `tank/zone/zion` est visible en lecture seule, que tous les descendants sont accessibles en écriture et que les jeux de données qui ne font pas partie de la hiérarchie parent sont invisibles. L'administrateur de zone ne peut pas modifier la propriété `sharenfs` car les zones non globales ne peuvent pas faire office de serveurs ZFS. L'administrateur de zone ne peut pas modifier la propriété `zoned` car cela entraînerait un risque de sécurité, tel que décrit dans la section suivante.

Les utilisateurs privilégiés dans la zone peuvent modifier toute autre propriété paramétrable, à l'exception des propriétés `quota` et `reservation`. Ce comportement permet à un administrateur de zone globale de contrôler l'espace disque occupé par tous les jeux de données utilisés par la zone non globale.

En outre, l'administrateur de zone globale ne peut pas modifier les propriétés `sharenfs` et `mountpoint` après la délégation d'un jeu de données à une zone non globale.

Explication de la propriété `zoned`

Lors qu'un jeu de données est délégué à une zone non globale, il doit être marqué spécialement pour que certaines propriétés ne soient pas interprétées dans le contexte de la zone globale. Lorsqu'un jeu de données est délégué à une zone non globale sous le contrôle d'un administrateur de zone, son contenu n'est plus de confiance. Comme dans tous les systèmes de fichiers, cela peut entraîner la présence de binaires `setuid`, de liens symboliques ou d'autres contenus douteux qui pourraient compromettre la sécurité de la zone globale. De plus, l'interprétation de la propriété `mountpoint` est impossible dans le contexte de la zone globale. Dans le cas contraire, l'administrateur de zone pourrait affecter l'espace de noms de la zone globale. Afin de résoudre ceci, ZFS utilise la propriété `zoned` pour indiquer qu'un jeu de données a été délégué à une zone non globale à un moment donné.

La propriété `zoned` est une valeur booléenne automatiquement activée lors de la première initialisation d'une zone contenant un jeu de données ZFS. L'activation manuelle de cette propriété par un administrateur de zone n'est pas nécessaire. Si la propriété `zoned` est définie, le

montage ou le partage du jeu de données est impossible dans la zone globale. Dans l'exemple suivant, le fichier `tank/zone/zion` a été délégué à une zone, alors que le fichier `tank/zone/global` ne l'a pas été :

```
# 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
```

Notez la différence entre la propriété `mountpoint` et le répertoire dans lequel le jeu de données `tank/zone/zion` est actuellement monté. La propriété `mountpoint` correspond à la propriété telle qu'elle est stockée dans le disque et non à l'emplacement auquel est monté le jeu de données sur le système.

Lors de la suppression d'un jeu de données d'une zone ou de la destruction d'une zone, la propriété `zoned` *n'est pas* effacée automatiquement. Ce comportement est dû aux risques de sécurité inhérents associés à ces tâches. Dans la mesure où un utilisateur qui n'est pas fiable dispose de l'accès complet au jeu de données et à ses enfants, la propriété `mountpoint` risque d'être configurée sur des valeurs erronées ou des binaires `setuid` peuvent exister dans les systèmes de fichiers.

Afin d'éviter tout risque de sécurité, l'administrateur global doit effacer manuellement la propriété `zoned` pour que le jeu de données puisse être utilisé à nouveau. Avant de configurer la propriété `zoned` sur `off`, assurez-vous que la propriété `mountpoint` du jeu de données et de tous ses enfants est configurée sur des valeurs raisonnables et qu'il n'existe aucun binaire `setuid`, ou désactivez la propriété `setuid`.

Après avoir vérifié qu'aucune vulnérabilité n'existe au niveau de la sécurité, vous pouvez désactiver la propriété `zoned` à l'aide de la commande `zfs set` ou `zfs inherit`. Si la propriété `zoned` est désactivée alors que le jeu de données est en cours d'utilisation au sein d'une zone, le système peut se comporter de façon imprévue. Ne modifiez la propriété que si vous êtes sûr que le jeu de données n'est plus en cours d'utilisation dans une zone non globale.

Utilisation de pools root ZFS de remplacement

Lors de sa création, un pool est intrinsèquement lié au système hôte. Le système hôte gère les informations du pool. Cela lui permet de détecter l'indisponibilité de ce dernier, le cas échéant. Même si elles sont utiles dans des conditions normales d'utilisation, ces informations peuvent causer des interférences lors de l'initialisation à partir d'autres médias ou lors de la création d'un pool sur un média amovible. La fonction de pool *root de remplacement* de ZFS permet de résoudre ce problème. Un pool root de remplacement n'est pas conservé d'une réinitialisation système à une autre et tous les points de montage sont modifiés de sorte à être relatifs au root du pool.

Création de pools root de remplacement ZFS

La création d'un pool root de remplacement s'effectue le plus souvent en vue d'une utilisation avec un média amovible. Dans ces circonstances, les utilisateurs souhaitent employer un système de fichiers unique et le monter à l'emplacement de leur choix dans le système cible. Lorsqu'un pool root de remplacement est créé à l'aide de l'option `zpool create -R`, le point de montage du système de fichiers root est automatiquement défini sur `/`, qui est l'équivalent du root de remplacement lui-même.

Dans l'exemple suivant, un pool nommé `morpheus` est créé à l'aide `/mnt` en tant que chemin root de remplacement :

```
# zpool create -R /mnt morpheus c0t0d0
# zfs list morpheus
NAME                  USED  AVAIL  REFER  MOUNTPOINT
morpheus             32.5K  33.5G     8K  /mnt
```

Notez le système de fichiers `morpheus` dont le point de montage est le root de remplacement du pool, `/mnt`. Le point de montage stocké sur le disque est `/` et le chemin complet de `/mnt` n'est interprété que dans le contexte du pool root de remplacement. Ce système de fichiers peut ensuite être exporté ou importé sous un pool root de remplacement arbitraire d'un autre système en respectant la syntaxe de *valeur du root secondaire* `-R`.

```
# zpool export morpheus
# zpool import morpheus
cannot mount '/': directory is not empty
# zpool export morpheus
# zpool import -R /mnt morpheus
# zfs list morpheus
NAME                  USED  AVAIL  REFER  MOUNTPOINT
morpheus             32.5K  33.5G     8K  /mnt
```

Importation de pools root de remplacement

L'importation de pool s'effectue également à l'aide d'un root de remplacement. Cette fonction permet de récupérer les données, le cas échéant, lorsque les points de montage ne doivent pas être interprétés dans le contexte du root actuel, mais sous un répertoire temporaire où pourront s'effectuer les réparations. Vous pouvez également utiliser cette fonction lors du montage de médias amovibles comme décrit dans la section précédente.

Dans l'exemple suivant, un pool nommé `morpheus` est importé à l'aide de `/mnt` en tant que chemin root de remplacement : Cet exemple part du principe que `morpheus` a été précédemment exporté.

```
# 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
```

◆ ◆ ◆ CHAPITRE 10

Dépannage d'Oracle Solaris ZFS et récupération de pool

Ce chapitre décrit les méthodes d'identification et de résolution des pannes de ZFS. Des informations relatives à la prévention des pannes sont également fournies.

Ce chapitre contient les sections suivantes :

- "Identification des problèmes ZFS" à la page 293
- "Résolution des problèmes matériels généraux" à la page 294
- "Identification des problèmes avec les pools de stockage ZFS" à la page 296
- "Résolution des problèmes des périphériques de stockage ZFS" à la page 301
- "Résolution des problèmes de système de fichiers ZFS" à la page 315
- "Réparation d'une configuration ZFS endommagée" à la page 325
- "Réparation d'un système impossible à réinitialiser" à la page 325

Pour plus d'informations sur la récupération complète du pool root, reportez-vous à la section "Restauration du pool root ZFS ou des instantanés du pool root" à la page 170.

Identification des problèmes ZFS

En tant que système de fichiers et gestionnaire de volumes combinés, ZFS peut rencontrer différentes pannes. Ce chapitre indique comment diagnostiquer les pannes matérielles générales et comment résoudre les problèmes liés au système de fichiers et au périphérique de pool. Vous pouvez rencontrer les types de problèmes suivants :

- **Problèmes matériels généraux** : les problèmes matériels peuvent affecter les performances de votre pool et la disponibilité des données de votre pool. Eliminez les problèmes matériels généraux, tels que la mémoire et les composants défectueux, avant de déterminer les problèmes à un niveau supérieur, tel qu'au niveau de vos pools et systèmes de fichiers.
- **Problèmes de pool de stockage ZFS**
  - "Identification des problèmes avec les pools de stockage ZFS" à la page 296
  - "Résolution des problèmes des périphériques de stockage ZFS" à la page 301

- “Résolution des problèmes de système de fichiers ZFS” à la page 315
- “Réparation d'une configuration ZFS endommagée” à la page 325
- “Réparation d'un système impossible à réinitialiser” à la page 325

Notez que les trois types d'erreurs peuvent se produire dans un même pool. Une procédure de réparation complète implique de trouver et de corriger une erreur, de passer à la suivante et ainsi de suite.

Résolution des problèmes matériels généraux

Consultez les sections suivantes pour déterminer si des problèmes de pool ou une indisponibilité de système de fichiers sont liés à un problème matériel, tel qu'une carte système, une mémoire, un périphérique, un HBA défectueux ou une erreur de configuration.

Par exemple, un disque défaillant ou défectueux situé sur un pool ZFS occupé peut considérablement dégrader les performances globales du système.

Si vous commencez par diagnostiquer et identifier les problèmes matériels, qui peuvent être plus faciles à détecter, et par effectuer une vérification matérielle, vous pouvez ensuite diagnostiquer les problèmes liés au pool et aux systèmes de fichiers comme décrit dans le reste de ce chapitre. Si vos configurations matérielles, de pool et de système de fichiers sont en bon état, pensez à diagnostiquer les problèmes d'application, qui sont généralement plus complexes à corriger et qui ne sont pas décrits dans ce guide.

Identification des pannes du matériel et des périphériques

Le gestionnaire des pannes Solaris recherche les problèmes logiciels, matériels et spécifiques aux périphériques en identifiant les informations de télémétrie des erreurs qui indiquent un symptôme spécifique dans un journal des erreurs puis signalent le diagnostic de panne réelle lorsque le symptôme d'erreur entraîne une panne réelle.

La commande suivante identifie toute panne logicielle ou matérielle.

```
# fmadm faulty
```

Exécutez régulièrement la commande ci-dessus pour identifier les services ou les périphériques défectueux.

Exécutez la commande suivante régulièrement pour identifier les erreurs liées au matériel ou aux périphériques.

```
# fmdump -eV | more
```

Les messages d'erreur de ce fichier journal qui décrivent les problèmes `vdev.open_failed`, `checksum` ou `io_failure` requièrent votre attention ou ils pourraient se transformer en pannes réelles affichées avec la commande faulty `fmadm`.

Si cette procédure indique qu'un périphérique est défectueux, c'est le bon moment pour vous assurer de la disponibilité d'un périphérique de remplacement.

Vous pouvez également suivre les autres erreurs de périphérique à l'aide de commande `iostat`. Utilisez la syntaxe suivante afin d'identifier un résumé des statistiques d'erreurs.

```
# 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
```

Dans la sortie ci-dessus, les erreurs sont signalées sur le disque interne `c2t0d0`. Utilisez la syntaxe suivante pour afficher des erreurs de périphérique plus détaillées.

```
# 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
```

Génération de rapports système de messages d'erreur ZFS

Outre le suivi permanent des erreur au sein du pool, ZFS affiche également des messages `syslog` lorsque des événements intéressants se produisent. Les scénarios suivants donnent lieu à des événements de notification :

- **Transition d'état de périphérique** : si l'état d'un périphérique devient `FAULTED`, ZFS consigne un message indiquant que la tolérance de pannes du pool risque d'être compromise. Un message similaire est envoyé si le périphérique est mis en ligne ultérieurement, restaurant la maintenance du pool.
- **Altération de données** : en cas de détection d'altération de données, ZFS consigne un message indiquant où et quand s'est produit la détection. Ce message n'est consigné que lors de la première détection. Les accès ultérieurs ne génèrent pas de message.
- **Défaillances de pool et de périphérique** : en cas de défaillance d'un pool ou d'un périphérique, le démon du gestionnaire de pannes rapporte ces erreurs par le biais de messages `syslog` et de la commande `fmdump`.

Si ZFS détecte un erreur de périphérique et la corrige automatiquement, aucune notification n'est générée. De telles erreurs ne constituent pas une défaillance de redondance de pool ou de l'intégrité des données. En outre, de telles erreurs sont typiquement dues à un problème de pilote accompagné de son propre jeu de messages d'erreur.

Identification des problèmes avec les pools de stockage ZFS

Les sections suivantes décrivent l'identification et la résolution des problèmes dans les systèmes de fichiers ZFS ou les pools de stockage :

- "Recherche de problèmes éventuels dans un pool de stockage ZFS" à la page 298
- "Consultation de la sortie de `zpool status`" à la page 298
- "Génération de rapports système de messages d'erreur ZFS" à la page 296

Les fonctions suivantes permettent d'identifier les problèmes au sein de la configuration ZFS :

- La commande `zpool status` permet d'afficher les informations détaillées des pools de stockage ZFS.
- Les défaillances de pool et de périphérique sont rapportées par le biais de messages de diagnostics ZFS/FMA.
- La commande `zpool history` permet d'afficher les commandes ZFS précédentes qui ont modifié les informations d'état de pool.

La commande `zpool status` permet de résoudre la plupart des problèmes au niveau de ZFS. Cette commande analyse les différentes erreurs système et identifie les problèmes les plus sévères. En outre, elle propose des actions à effectuer et un lien vers un article de connaissances

pour de plus amples informations. Notez que cette commande n'identifie qu'un seul problème dans le pool, même si plusieurs problèmes existent. Par exemple, les erreurs d'altération de données sont généralement provoquées par la panne d'un périphérique, mais le remplacement d'un périphérique défaillant peut ne pas résoudre tous les problèmes d'altération de données.

En outre, un moteur de diagnostic ZFS diagnostique et signale les défaillances au niveau du pool et des périphériques. Les erreurs liées aux sommes de contrôle, aux E/S, aux périphériques et aux pools font également l'objet d'un rapport lorsqu'elles sont liées à ces défaillances. Les défaillances ZFS telles que rapportées par `fmd` s'affichent sur la console ainsi que les dans le fichier de messages système. Dans la plupart des cas, le message `fmd` vous dirige vers la commande `zpool status` pour obtenir des instructions supplémentaires de récupération.

Le processus de récupération est comme décrit ci-après :

- Le cas échéant, la commande `zpool history` permet d'identifier les commandes ZFS ayant précédé le scénario d'erreur. Par exemple :

  ```
  # 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
  ```

  Dans cette sortie, notez que les sommes de contrôle sont désactivées pour le système de fichiers `tank/eric`. Cette configuration est déconseillée.
- Identifiez les erreurs à l'aide des messages `fmd` affichés sur la console système ou dans le fichier `/var/adm/messages`.
- Obtenez des instructions de réparation supplémentaires grâce à la commande `zpool status -x`.
- Réparez les pannes. Pour ce faire, suivez les étapes ci-après :
  - Remplacez le périphérique indisponible ou manquant et mettez-le en ligne.
  - Restaurez la configuration défaillante ou les données altérées à partir d'une sauvegarde.
  - Vérifiez la récupération à l'aide de la commande `zpool status -x`.
  - Sauvegardez la configuration restaurée, le cas échéant.

Cette section explique comment interpréter la sortie `zpool status` afin de diagnostiquer le type de défaillances pouvant survenir. Même si la commande effectue automatiquement le travail, vous devez comprendre exactement les problèmes identifiés pour diagnostiquer la panne. Les sections suivantes expliquent comment corriger les différents types de problèmes que vous risquez de rencontrer.

Recherche de problèmes éventuels dans un pool de stockage ZFS

La méthode la plus simple pour déterminer s'il existe des problèmes connus sur le système consiste à exécuter la commande `zpool status -x`. Cette commande décrit uniquement les pools présentant des problèmes. Si tous les pools du système fonctionnent correctement, la commande affiche les éléments suivants :

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

Sans l'indicateur `-x`, la commande affiche l'état complet de tous les pools (ou du pool demandé s'il est spécifié sur la ligne de commande), même si les pools sont autrement fonctionnels.

Pour de plus amples informations sur les options de ligne de commande de la commande `zpool status`, reportez-vous à la section "Requête d'état de pool de stockage ZFS" à la page 88.

Consultation de la sortie de `zpool status`

La sortie complète de `zpool status` est similaire à ce qui suit :

```
# 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
```

Cette sortie est décrite dans la section suivante.

Informations globales d'état des pools

Cette section de la sortie `zpool status` se compose des champs suivants, certains d'entre eux n'étant affichés que pour les pools présentant des problèmes :

`pool` Identifie le nom du pool.

`state` Indique l'état de maintenance actuel du pool. Ces informations concernent uniquement la capacité de pool à fournir le niveau de réplication requis.

`status` Décrit les problèmes du pool. Ce champ est absent si aucune erreur n'est détectée.

`action` Action recommandée pour la réparation des erreurs. Ce champ est absent si aucune erreur n'est détectée.

`see` Fait référence à un article de connaissances contenant des informations de réparation détaillées. Les articles en ligne sont mis à jour plus régulièrement que ce guide. Par conséquent, vous devez vous y reporter pour obtenir les procédures de réparation les plus récentes. Ce champ est absent si aucune erreur n'est détectée.

`scrub` Identifie l'état actuel d'une opération de nettoyage. Ce champ peut indiquer la date et l'heure du dernier nettoyage, un nettoyage en cours ou l'absence de demande de nettoyage.

`errors` Identifie les erreurs de données ou l'absence d'erreurs de données connues.

Informations sur la configuration des pools de stockage ZFS

Le champ `config` de la sortie `zpool status` décrit la configuration des périphériques inclus dans le pool, ainsi que leur état et toute erreur générée à partir des périphériques. L'état peut être l'un des suivants : `ONLINE`, `FAULTED`, `DEGRADED` ou `SUSPENDED`. Si l'état n'est pas `ONLINE`, la tolérance de pannes du pool a été compromise.

La deuxième section de la sortie de configuration affiche des statistiques d'erreurs. Ces erreurs se divisent en trois catégories :

- `READ` : erreurs d'E/S qui se sont produites lors de l'envoi d'une demande de lecture
- `WRITE` : erreurs d'E/S qui se sont produites lors de l'envoi d'une demande d'écriture
- `CKSUM` : erreurs de somme de contrôle signifiant que le périphérique a renvoyé des données altérées en réponse à une demande de lecture.

Il est possible d'utiliser ces erreurs pour déterminer si les dommages sont permanents. Des erreurs d'E/S peu nombreuses peuvent indiquer une interruption de service temporaire. Si elles sont nombreuses, il est possible que le périphérique présente un problème permanent. Ces erreurs ne correspondent pas nécessairement à l'altération de données telle qu'interprétée par les applications. Si la configuration du périphérique est redondante, les périphériques peuvent présenter des erreurs impossibles à corriger, même si aucune erreur ne s'affiche au niveau du périphérique RAID-Z ou du miroir. Dans ce cas, ZFS a récupéré les données correctes et a réussi à réparer les données endommagées à partir des répliques existantes.

Pour de plus amples informations sur l'interprétation de ces erreurs, reportez-vous à la section “Détermination du type de panne de périphérique” à la page 305.

Enfin, les informations auxiliaires supplémentaire sont affichées dans la dernière colonne de la sortie de `zpool status`. Ces informations s'étendent dans le champ `state` et facilitent le diagnostic des pannes. Si l'état d'un périphérique est `UNAVAIL`, ce champ indique si le

périphérique est inaccessible ou si les données du périphérique sont altérées. Si le périphérique est en cours de réargenture, ce champ affiche la progression du processus.

Pour de plus amples informations sur la surveillance de la progression de la réargenture, reportez-vous à la section "Affichage de l'état de réargenture" à la page 313.

Etat du nettoyage des pools de stockage ZFS

La section sur le nettoyage de la sortie `zpool status` décrit l'état actuel de toute opération de nettoyage explicite. Ces informations sont distinctes de la détection d'erreurs dans le système, mais il est possible de les utiliser pour déterminer l'exactitude du rapport d'erreurs d'altération de données. Si le dernier nettoyage s'est récemment terminé, toute altération de données existante aura probablement déjà été détecté.

Les messages de fin de nettoyage subsistent après plusieurs réinitialisations du système.

Pour de plus amples informations sur le nettoyage de données et l'interprétation de ces informations, reportez-vous à la section "Contrôle de l'intégrité d'un système de fichiers ZFS" à la page 315.

Erreurs d'altération de données ZFS

La commande `zpool status` indique également si des erreurs connues sont associées au pool. La détection de ces erreurs a pu s'effectuer lors du nettoyage des données ou lors des opérations normales. Le système de fichiers ZFS gère un journal persistant de toutes les erreurs de données associées à un pool. Ce journal tourne à chaque fois qu'un nettoyage complet du système est terminé.

Les erreurs d'altération de données constituent toujours des erreurs fatales. Elles indiquent une erreur d'E/S dans au moins une application, en raison de la présence de données altérées au sein du pool. Les erreurs de périphérique dans un pool redondant n'entraînent pas d'altération de données et ne sont pas enregistrées en tant que partie de ce journal. Par défaut, seul le nombre d'erreurs trouvées s'affiche. Vous pouvez obtenir la liste complète des erreurs et de leurs spécificités à l'aide de l'option `zpool status -v`. Par exemple :

```
# 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
```

La commande `fmd` affiche également un message similaire dans la console système et le fichier `/var/adm/messages`. La commande `fmdump` permet également de réaliser le suivi de ces messages.

Pour de plus amples informations sur l'interprétation d'erreurs d'altération de données, reportez-vous à la section "Identification du type d'altération de données" à la page 320.

Résolution des problèmes des périphériques de stockage ZFS

Consultez les sections suivantes pour résoudre un problème de périphérique manquant, supprimé ou défectueux.

Résolution d'un périphérique manquant ou supprimé

Si l'ouverture d'un périphérique est impossible, ce dernier s'affiche dans l'état `UNAVAIL` dans la sortie de `zpool status`. Cet état indique que ZFS n'a pas pu ouvrir le périphérique lors du premier accès au pool ou que le périphérique est devenu indisponible par la suite. Si le périphérique rend un périphérique de niveau supérieur indisponible, l'intégralité du pool devient inaccessible. Dans le cas contraire, la tolérance de pannes du pool risque d'être compromise. Quel que soit le cas, le périphérique doit simplement être reconnecté au système pour fonctionner à nouveau normalement. Si vous devez remplacer un périphérique qui est `UNAVAIL` car il est défectueux, reportez-vous à la section "Remplacement d'un périphérique dans un pool de stockage ZFS" à la page 307.

Si l'état d'un périphérique est `UNAVAIL` dans un pool root ou dans un pool root mis en miroir, consultez les références suivantes :

- **La mise en miroir du disque de pool root a échoué** – "Initialisation à partir d'un disque alternatif d'un pool root ZFS mis en miroir" à la page 163
- **Remplacement d'un disque dans un pool root**
  - "Remplacement d'un disque dans le pool root ZFS" à la page 170
- **Reprise sur sinistre de pool root complet** : "Restauration du pool root ZFS ou des instantanés du pool root" à la page 170

Par exemple, après une panne de périphérique, `fmd` peut afficher un message similaire au suivant :

```
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.
```

Pour afficher des informations détaillées sur le problème du périphérique et sa résolution, utilisez la commande `zpool status -x`. Par exemple :

```
# 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
```

Cette sortie indique que le périphérique `c2t1d0` ne fonctionne pas. Si vous estimez que le périphérique est défectueux, remplacez-le.

Si nécessaire, exécutez ensuite la commande `zpool online` pour mettre le périphérique remplacé en ligne. Par exemple :

```
# zpool online tank c2t1d0
```

Signalez à FMA que le périphérique a été remplacé si la sortie de `fmadm faulty` identifie l'erreur de périphérique. Par exemple :

```
# 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
```

Confirmez ensuite que le pool dont le périphérique a été remplacé fonctionne correctement. Par exemple :

```
# zpool status -x tank
pool 'tank' is healthy
```

Résolution d'un périphérique supprimé

Si un périphérique est entièrement supprimé du système, ZFS s'assure que le périphérique ne peut pas être ouvert et il le place dans l'état `REMOVED`. En fonction du niveau de réplication des données du pool, ce retrait peut résulter ou non en une indisponibilité de la totalité du pool. Le pool reste accessible en cas de suppression d'un périphérique mis en miroir ou RAID-Z. Un pool peut renvoyer l'état `UNAVAIL`. Cela signifie qu'aucune donnée n'est accessible jusqu'à ce que le périphérique soit reconnecté selon les conditions suivantes :

- Si tous les composants d'un miroir sont supprimés
- Si plusieurs périphériques d'un périphérique RAID-Z (`raidz1`) sont supprimés
- Si le périphérique de niveau supérieur est supprimé dans une configuration contenant un seul disque

Reconnexion physique d'un périphérique

La reconnexion d'un périphérique dépend du périphérique en question. S'il s'agit d'un disque connecté au réseau, la connectivité au réseau doit être restaurée. S'il s'agit d'un périphérique USB ou autre média amovible, il doit être reconnecté au système. S'il s'agit d'un disque local, un contrôleur est peut-être tombé en panne, rendant le périphérique invisible au système. Dans ce cas, il faut remplacer le contrôleur pour que les disques soient à nouveau disponibles. D'autres problèmes existent et dépendent du type de matériel et de sa configuration. Si un disque tombe en panne et n'est plus visible pour le système, le périphérique doit être traité comme un périphérique endommagé. Suivez les procédures décrites dans la section "Remplacement ou réparation d'un périphérique endommagé " à la page 304.

Un pool peut être SUSPENDED si la connectivité du périphérique est problématique. Un pool SUSPENDED reste en état wait jusqu'à ce que le problème du périphérique soit résolu. Par exemple :

```
# 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
```

Une fois la connectivité de périphérique restaurée, effacez les erreurs de pool ou de périphérique.

```
# zpool clear cybermen
# fmadm repaired zfs://pool=name/vdev=guid
```

Notification relative à la disponibilité de périphériques dans ZFS

Une fois le périphérique reconnecté au système, sa disponibilité peut être détectée automatiquement ou non dans ZFS. Si le pool était précédemment UNAVAIL ou SUSPENDED, ou si le système a été réinitialisé en tant que partie de la procédure attach, ZFS rebalaye automatiquement tous les périphériques lors de la tentative d'ouverture du pool. Si le pool était endommagé et que le périphérique a été remplacé alors que le système était en cours d'exécution, vous devez indiquer à ZFS que le périphérique est dorénavant disponible et qu'il est prêt à être rouvert à l'aide de la commande zpool online. Par exemple :

```
# zpool online tank c0t1d0
```

Pour de plus amples informations sur la remise en ligne de périphériques, reportez-vous à la section "Mise en ligne d'un périphérique" à la page 76.

Remplacement ou réparation d'un périphérique endommagé

Cette section explique comment déterminer les types de panne de périphériques, effacer les erreurs transitoires et remplacer un périphérique.

Détermination du type de panne de périphérique

L'expression *périphérique endommagé* peut décrire un grand nombre de situations :

- **"Bit rot" ou dégradation progressive des données** : sur la durée, des événements aléatoires, tels que les influences magnétiques et les rayons cosmiques, peuvent entraîner une inversion des bits stockés dans le disque. Ces événements sont relativement rares mais, cependant, assez courants pour entraîner des altérations de données potentielles dans des systèmes de grande taille ou de longue durée.
- **Lectures ou écritures mal dirigées** : les bogues de microprogrammes ou les pannes de matériel peuvent entraîner un référencement incorrect de l'emplacement du disque par des lectures ou écritures de blocs entiers. Ces erreurs sont généralement transitoires, mais un grand nombre d'entre elles peut indiquer un disque défectueux.
- **Erreur d'administrateur** : les administrateurs peuvent écraser par erreur des parties du disque avec des données erronées (la copie de `/dev/zero` sur des parties du disque, par exemple) qui entraînent la détérioration permanente du disque. Ces erreurs sont toujours transitoires.
- **Interruption temporaire de service** : un disque peut être temporairement indisponible, entraînant l'échec des E/S. En général, cette situation est associée aux périphériques connectés au réseau, mais les disques locaux peuvent également connaître des interruptions temporaires de service. Ces erreurs peuvent être transitoires ou non.
- **Matériel défectueux ou peu fiable** : cette situation englobe tous les problèmes liés à un matériel défectueux, y compris les erreurs d'E/S cohérentes, les transports défectueux entraînant des altérations aléatoires ou des pannes. Ces erreurs sont typiquement permanentes.
- **Périphérique mis hors ligne** : si un périphérique est hors ligne, il est considéré comme ayant été mis hors ligne par l'administrateur, parce qu'il était défectueux. L'administrateur qui a mis ce dispositif hors ligne peut déterminer si cette hypothèse est exacte.

Il est parfois difficile de déterminer la nature exacte de la panne du dispositif. La première étape consiste à examiner le décompte d'erreurs dans la sortie de `zpool status`. Par exemple :

```
# 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
```

Les erreurs sont divisées en erreurs d'E/S et en erreurs de sommes de contrôle. Ces deux catégories peuvent indiquer le type de panne possible. Une opération typique renvoie un très petit nombre d'erreurs (quelques-unes sur une longue période). Si les erreurs sont nombreuses, un périphérique est probablement en panne ou sur le point de tomber en panne. Cependant, une erreur provoquée par un administrateur peut également entraîner un grand nombre d'erreurs. Le journal système `syslog` constitue une autre source d'informations. Si le journal présente un grand nombre de messages SCSI ou de pilote Fibre Channel, il existe probablement de graves problèmes matériels. L'absence de messages `syslog` indique que les dommages sont probablement transitoires.

L'objectif est de répondre à la question suivante :

Est-il possible qu'une autre erreur se produise dans ce périphérique ?

Les erreurs qui ne se produisent qu'une fois sont considérées *transitoires* et n'indiquent pas une panne potentielle. Les erreurs suffisamment persistantes ou sévères pour indiquer une panne matérielle potentielle sont considérées comme étant des *erreurs fatales*. Aucun logiciel automatisé actuellement disponible avec ZFS ne permet de déterminer le type d'erreur. Par conséquent, l'administrateur doit procéder manuellement. Une fois l'erreur déterminée, vous pouvez réaliser l'action adéquate. En cas d'erreurs fatales, effacez les erreurs transitoires ou remplacez le périphérique. Ces procédures de réparation sont décrites dans les sections suivantes.

Même si les erreurs de périphériques sont considérées comme étant transitoires, elles peuvent tout de même entraîner des erreurs de données impossibles à corriger au sein du pool. Ces erreurs requièrent des procédures de réparation spéciales, même si le périphérique sous-jacent est considéré comme étant fonctionnel ou réparé. Pour de plus amples informations sur la réparation d'erreurs de données, reportez-vous à la section “Réparation de données endommagées” à la page 320.

Suppression des erreurs de périphérique transitoires

Si les erreurs de périphérique sont considérées comme étant transitoires, dans la mesure où il est peu probable qu'elles affectent la maintenance du périphérique, elles peuvent être effacées en toute sécurité pour indiquer qu'aucune erreur fatale ne s'est produite. Pour effacer les compteurs d'erreurs pour les périphériques mis en miroir ou RAID-Z, utilisez la commande `zpool clear`. Par exemple :

```
# zpool clear tank c1t1d0
```

Cette syntaxe efface toutes les erreurs du périphérique et tout décompte d'erreurs de données associées au périphérique.

Pour effacer toutes les erreurs associées aux périphériques virtuels du pool et tout décompte d'erreurs de données associées au pool, respectez la syntaxe suivante :

```
# zpool clear tank
```

Pour de plus amples informations relatives à la suppression des erreurs de pool, reportez-vous à la section “Effacement des erreurs de périphérique de pool de stockage” à la page 77.

Remplacement d'un périphérique dans un pool de stockage ZFS

Si le périphérique présente ou risque de présenter une panne permanente, il doit être remplacé. Le remplacement du périphérique dépend de la configuration.

- “Détermination de la possibilité de remplacement du périphérique” à la page 307
- “Périphériques impossibles à remplacer” à la page 308
- “Remplacement d'un périphérique dans un pool de stockage ZFS” à la page 308
- “Affichage de l'état de réargenture” à la page 313

Détermination de la possibilité de remplacement du périphérique

Si le périphérique à remplacer fait partie d'une configuration redondante, il doit exister suffisamment de répliques pour permettre la récupération des données correctes. Si deux disques d'un miroir à quatre directions sont `UNAVAIL`, chaque disque peut être remplacé car des répliques saines sont disponibles. Cependant, si deux disques dans un périphérique virtuel RAID-Z à quatre directions (`raidz1`) sont `UNAVAIL`, aucun disque ne peut être remplacé en l'absence de répliques suffisantes permettant de récupérer les données. Si le périphérique est endommagé mais en ligne, il peut être remplacé tant que l'état du pool n'est pas `UNAVAIL`. Toutefois, toute donnée endommagée sur le périphérique est copiée sur le nouveau périphérique, à moins que le nombre de copies des données non endommagées soit déjà suffisant.

Dans la configuration suivante, le disque `c1t1d0` peut être remplacé et toute donnée du pool est copiée à partir de la réplique saine, `c1t0d0`.

```
    mirror            DEGRADED
    c1t0d0             ONLINE
    c1t1d0             FAULTED
```

Le disque `c1t0d0` peut également être remplacé, mais un autorétablissement des données est impossible, car il n'existe aucune réplique correcte.

Dans la configuration suivante, aucun des disques `UNAVAIL` ne peut être remplacé. Les disques `ONLINE` ne peuvent pas l'être non plus, car le pool lui-même est `UNAVAIL`.

```
    raidz               FAULTED
    c1t0d0              ONLINE
    c2t0d0              FAULTED
    c3t0d0              FAULTED
    c4t0d0              ONLINE
```

Dans la configuration suivante, chacun des disques de niveau supérieur peut être remplacé. Cependant, les données incorrectes seront également copiées dans le nouveau disque, le cas échéant.

```
c1t0d0          ONLINE
c1t1d0          ONLINE
```

Si les deux disques sont `UNAVAIL`, tout remplacement est impossible car le pool lui-même est `UNAVAIL`.

Périphériques impossibles à remplacer

Si la perte d'un périphérique rend un pool `UNAVAIL` ou si le périphérique contient trop d'erreurs de données dans une configuration non redondante, le remplacement du périphérique en toute sécurité est impossible. En l'absence de redondance suffisante, il n'existe pas de données correctes avec lesquelles réparer le périphérique défectueux. Dans ce cas, la seule option est de détruire le pool, recréer la configuration et restaurer les données à partir d'une copie de sauvegarde.

Pour de plus amples informations sur la restauration d'un pool entier, reportez-vous à la section "Réparation de dommages présents dans l'ensemble du pool de stockage ZFS" à la page 323.

Remplacement d'un périphérique dans un pool de stockage ZFS

Après avoir déterminé qu'il est possible de remplacer un périphérique, exécutez la commande `zpool replace` pour le remplacer effectivement. Exécutez la commande suivante si vous remplacez le périphérique endommagé par un autre périphérique différent :

```
# zpool replace tank c1t1d0 c2t0d0
```

Cette commande lance la migration de données vers le nouveau périphérique, soit à partir du périphérique endommagé, soit à partir d'autres périphériques du pool s'il s'agit d'une configuration redondante. Une fois l'exécution de la commande terminée, le périphérique endommagé est séparé de la configuration. Il peut dorénavant être retiré du système. Si vous avez déjà retiré le périphérique et que vous l'avez remplacé par un autre dans le même emplacement, utilisez la forme "périphérique unique" de la commande. Par exemple :

```
# zpool replace tank c1t1d0
```

Cette commande formate adéquatement un disque non formaté et resynchronise ensuite les données à partir du reste de la configuration.

Pour de plus amples informations sur la commande `zpool replace` reportez-vous à la section "Remplacement de périphériques dans un pool de stockage" à la page 77.

EXEMPLE 10–1 Remplacement d'un disque SATA dans un pool de stockage ZFS

L'exemple suivant illustre le remplacement d'un périphérique (`c1t3d0`) du pool de stockage mis en miroir `tank` sur un système équipé de périphériques SATA. Pour remplacer le disque `c1t3d0` par un nouveau au même emplacement (`c1t3d0`), annulez la configuration du disque avant de procéder au remplacement. Si le disque qui doit être remplacé n'est pas un disque SATA, reportez-vous à la section "Remplacement de périphériques dans un pool de stockage" à la page 77.

Voici les principales étapes à suivre :

- Déconnectez le disque (`c1t3d0`) à remplacer. Vous ne pouvez pas annuler la configuration d'un disque SATA utilisé.
- Utilisez la commande `cfgadm` pour identifier le disque SATA (`c1t3d0`) dont la configuration doit être annulée et annulez-la. Dans cette configuration en miroir, le pool est endommagé et le disque est hors ligne, mais le pool reste disponible.
- Remplacez le disque (`c1t3d0`). Vérifiez que la DEL bleue `Ready to Remove`, indiquant que le périphérique est prêt à être retiré, est allumée avant de retirer le lecteur `UNAVAIL`, si disponible.
- Reconfigurez le disque SATA (`c1t3d0`).
- Mettez le nouveau disque (`c1t3d0`) en ligne.
- Exécutez la commande `zpool replace` pour remplacer le disque (`c1t3d0`).

Remarque – Si vous avez précédemment défini la propriété de pool `autoreplace` sur `on`, tout nouveau périphérique détecté au même emplacement physique qu'un périphérique appartenant précédemment au pool est automatiquement formaté et remplacé sans recourir à la commande `zpool replace`. Cette fonction n'est pas prise en charge sur tous les types de matériel.

- Si un disque défectueux est automatiquement remplacé par un disque hot spare, vous devrez peut-être déconnecter le disque hot spare une fois le disque défectueux remplacé. Par exemple, si `c2t4d0` reste actif comme disque hot spare actif une fois le disque défectueux remplacé, déconnectez-le.

```
# zpool detach tank c2t4d0
```

- Si FMA signale le périphérique défaillant, effacez la panne de périphérique.

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

EXEMPLE 10–1 Remplacement d'un disque SATA dans un pool de stockage ZFS *(Suite)*

L'exemple suivant explique étape par étape comment remplacer un disque dans un pool de stockage ZFS.

```
# zpool offline tank c1t3d0
# cfgadm | grep c1t3d0
sata1/3::dsk/c1t3d0            disk         connected    configured   ok
# cfgadm -c unconfigure sata1/3
Unconfigure the device at: /devices/pci@0,0/pci1022,7458@2/pci11ab,11ab@1:3
This operation will suspend activity on the SATA device
Continue (yes/no)? yes
# cfgadm | grep sata1/3
sata1/3                        disk         connected    unconfigured ok
<Physically replace the failed disk c1t3d0>
# cfgadm -c configure sata1/3
# cfgadm | grep sata1/3
sata1/3::dsk/c1t3d0            disk         connected    configured   ok
# zpool online tank c1t3d0
# zpool replace tank c1t3d0
# zpool status tank
  pool: tank
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:17:32 2010
config:

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

errors: No known data errors
```

Notez que la commande `zpool output` affiche parfois l'ancien disque et le nouveau sous l'en-tête de *remplacement*. Par exemple :

```
replacing     DEGRADED     0     0    0
  c1t3d0s0/o  FAULTED      0     0    0
  c1t3d0      ONLINE       0     0    0
```

Ce texte signifie que la procédure de remplacement et la réargenture du nouveau disque sont en cours.

Pour remplacer un disque (`c1t3d0`) par un autre disque (`c4t3d0`), il suffit d'exécuter la commande `zpool replace`. Par exemple :

EXEMPLE 10–1 Remplacement d'un disque SATA dans un pool de stockage ZFS *(Suite)*

```
# zpool replace tank c1t3d0 c4t3d0
# zpool status
  pool: tank
 state: DEGRADED
 scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:

        NAME             STATE     READ WRITE CKSUM
        tank             DEGRADED     0     0     0
          mirror-0       ONLINE       0     0     0
            c0t1d0       ONLINE       0     0     0
            c1t1d0       ONLINE       0     0     0
          mirror-1       ONLINE       0     0     0
            c0t2d0       ONLINE       0     0     0
            c1t2d0       ONLINE       0     0     0
          mirror-2       DEGRADED     0     0     0
            c0t3d0       ONLINE       0     0     0
            replacing    DEGRADED     0     0     0
              c1t3d0     OFFLINE      0     0     0
              c4t3d0     ONLINE       0     0     0

errors: No known data errors
```

La commande `zpool status` doit parfois être exécutée plusieurs fois jusqu'à la fin du remplacement du disque.

```
# 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
```

EXEMPLE 10–2 Remplacement d'un périphérique de journalisation défaillant

ZFS identifie les défaillances de journal d'intention dans la sortie de commande `zpool status`. Le composant FMA (Fault Management Architecture) signale également ces erreurs. ZFS et FMA décrivent comment récupérer les données en cas de défaillance du journal d'intention.

L'exemple suivant montre comment récupérer les données d'un périphérique de journalisation défaillant (`c0t5d0`) dans le pool de stockage (`pool`). Voici les principales étapes à suivre :

EXEMPLE 10–2 Remplacement d'un périphérique de journalisation défaillant *(Suite)*

- Vérifiez la sortie `zpool status -x` et le message de diagnostic FMA, décrits ici :

  `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&alias=EVENT:ZFS-8000-K4`
- Remplacez physiquement le périphérique de journalisation défaillant.
- Mettez le nouveau périphérique de journalisation en ligne.
- Effacez la condition d'erreur du pool.
- Effacez l'erreur FMA.

```
# zpool status -x
  pool: pool
 state: FAULTED
status: One or more of the intent logs could not be read.
        Waiting for adminstrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
        or ignore the intent log records by running 'zpool clear'.
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        pool        FAULTED      0     0     0 bad intent log
          mirror    ONLINE       0     0     0
            c0t1d0  ONLINE       0     0     0
            c0t4d0  ONLINE       0     0     0
        logs        FAULTED      0     0     0 bad intent log
          c0t5d0    UNAVAIL      0     0     0 cannot open
<Physically replace the failed log device>
# zpool online pool c0t5d0
# zpool clear pool
```

Par exemple, si le système s'arrête soudainement avant que les opérations d'écriture synchrone ne soient affectées à un pool disposant d'un périphérique de journalisation distinct, un message tel que le suivant s'affiche :

```
# 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
```

EXEMPLE 10–2 Remplacement d'un périphérique de journalisation défaillant *(Suite)*

```
        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
```

Vous pouvez résoudre la panne du périphérique de journalisation de l'une des façons suivantes :

- Remplacez ou récupérez le périphérique de journalisation. Dans cet exemple, le périphérique de journalisation est `c0t5d0`.
- Mettez le périphérique de journalisation en ligne.

  ```
  # zpool online pool c0t5d0
  ```
- Réinitialisez la condition d'erreur de périphérique de journalisation défaillante.

  ```
  # zpool clear pool
  ```

Pour effectuer une récupération suite à cette erreur sans remplacer le périphérique de journalisation défaillant, vous pouvez effacer l'erreur à l'aide de la commande `zpool clear`. Dans ce scénario, le pool fonctionnera en mode dégradé et les enregistrements de journal seront enregistrés dans le pool principal jusqu'à ce que le périphérique de journalisation distinct soit remplacé.

Envisagez d'utiliser des périphériques de journalisation mis en miroir afin d'éviter un scénario de défaillance de périphérique de journalisation.

Affichage de l'état de réargenture

Le processus de remplacement d'un périphérique peut prendre beaucoup de temps, selon la taille du périphérique et la quantité de données dans le pool. Le processus de déplacement de données d'un périphérique à un autre s'appelle la *réargenture*. Vous pouvez la surveiller à l'aide de la commande `zpool status`.

Les systèmes de fichiers traditionnels effectuent la réargenture de données au niveau du bloc. Dans la mesure où ZFS élimine la séparation en couches artificielles du gestionnaire de volume, il peut effectuer la réargenture de façon bien plus puissante et contrôlée. Les deux avantages de cette fonction sont comme suite :

- ZFS n'effectue la réargenture que de la quantité minimale de données requises. Dans le cas d'une brève interruption de service (par rapport à un remplacement complet d'un périphérique), vous pouvez effectuer la réargenture du disque en quelques minutes ou quelques secondes. Lors du remplacement d'un disque entier, la durée du processus de

réargenture est proportionnelle à la quantité de données utilisées dans le disque. Le remplacement d'un disque de 500 Go ne dure que quelques secondes si le pool ne contient que quelques giga-octets d'espace utilisé.

- La réargenture est un processus fiable qui peut être interrompu, le cas échéant. En cas de mise hors-tension ou de réinitialisation du système, le processus de réargenture reprend exactement là où il s'est arrêté, sans requérir une intervention manuelle.

La commande `zpool status` permet de visualiser le processus de réargenture. Par exemple :

```
# 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
```

Dans cet exemple, le disque `c1t0d0` est remplacé par `c2t0d0`. Cet événement est observé dans la sortie d'état par la présence du périphérique virtuel `replacing` de la configuration. Ce périphérique n'est pas réel et ne permet pas de créer un pool. L'objectif de ce périphérique consiste uniquement à afficher le processus de réargenture et à identifier le périphérique en cours de remplacement.

Notez que tout pool en cours de réargenture se voit attribuer l'état `ONLINE` ou `DEGRADED` car il ne peut pas fournir le niveau souhaité de redondance tant que le processus n'est pas terminé. La réargenture s'effectue aussi rapidement que possible, mais les E/S sont toujours programmées avec une priorité inférieure à celle des E/S requises par l'utilisateur afin de minimiser l'impact sur le système. Une fois la réargenture terminée, la nouvelle configuration complète s'applique, remplaçant l'ancienne configuration. Par exemple :

```
# 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
```

L'état du pool est à nouveau `ONLINE` et le disque défectueux d'origine (`c1t0d0`) a été supprimé de la configuration.

Résolution des problèmes de système de fichiers ZFS

Résolution des problèmes de données dans un pool de stockage ZFS

Parmi les exemples de problèmes de données, on trouve :

- Erreurs d'E/S transitoires causées par un disque ou un contrôleur défaillant
- Altération de données sur disque causée par les rayons cosmiques
- Bogues de pilotes entraînant des transferts de données vers ou à partir d'un emplacement erroné
- Ecrasement accidentel de parties du périphérique physique par un utilisateur

Certaines erreurs sont transitoires, par exemple une erreur d'E/S aléatoire alors que le contrôleur rencontre des problèmes. Dans d'autres cas, les dommages sont permanents, par exemple lors de la détérioration sur disque. En outre, même si les dommages sont permanents, cela ne signifie pas que l'erreur est susceptible de se reproduire. Par exemple, si un utilisateur écrase une partie d'un disque par accident, aucune panne matérielle ne s'est produite et il est inutile de remplacer le périphérique. L'identification du problème exact dans un périphérique n'est pas une tâche aisée. Elle est abordée plus en détail dans une section ultérieure.

Contrôle de l'intégrité d'un système de fichiers ZFS

Il n'existe pas d'utilitaire `fsck` équivalent pour ZFS. Cet utilitaire remplissait deux fonctions : réparer et valider le système de fichiers.

Réparation du système de fichiers

Avec les systèmes de fichiers classiques, la méthode d'écriture des données est affectée par les pannes inattendues entraînant des incohérences de systèmes de fichiers. Un système de fichiers classique n'étant pas transactionnel, les blocs non référencés, les comptes de liens défectueux ou autres structures de systèmes de fichiers incohérentes sont possibles. L'ajout de la journalisation résout certains de ces problèmes, mais peut entraîner des problèmes supplémentaires lorsque la restauration du journal est impossible. Une incohérence des données sur disque dans une

configuration ZFS ne se produit qu'à la suite d'une panne de matérielle (auquel cas le pool aurait dû être redondant) ou en présence d'un bogue dans le logiciel ZFS.

L'utilitaire `fsck` répare les problèmes connus spécifiques aux systèmes de fichiers UFS. La plupart des problèmes au niveau des pools de stockage ZFS sont généralement liés à un matériel défaillant ou à des pannes de courant. En utilisant des pools redondants, vous pouvez éviter de nombreux problèmes. Si le pool est endommagé suite à une défaillance de matériel ou à une coupure de courant, reportez-vous à la section "Réparation de dommages présents dans l'ensemble du pool de stockage ZFS" à la page 323.

Si le pool n'est pas redondant, le risque qu'une altération de système de fichiers puisse rendre tout ou partie de vos données inaccessibles est toujours présent.

Validation du système de fichiers

Outre la réparation du système de fichiers, l'utilitaire `fsck` valide l'absence de problème relatif aux données sur le disque. Cette tâche requiert habituellement le démontage du système de fichiers et en l'exécution de l'utilitaire `fsck`, éventuellement en mettant le système en mode utilisateur unique lors du processus. Ce scénario entraîne une indisponibilité proportionnelle à la taille du système de fichiers en cours de vérification. Plutôt que de requérir un utilitaire explicite pour effectuer la vérification nécessaire, ZFS fournit un mécanisme pour effectuer une vérification de routine des incohérences. Cette fonctionnalité, appelée *nettoyage*, est fréquemment utilisée dans les systèmes de mémoire et autres systèmes comme méthode de détection et de prévention d'erreurs pour éviter qu'elles entraînent des pannes matérielles ou logicielles.

Contrôle du nettoyage de données ZFS

Si ZFS rencontre une erreur, soit via le nettoyage ou lors de l'accès à un fichier à la demande, l'erreur est journalisée en interne pour vous donner une vue d'ensemble rapide de toutes les erreurs connues au sein du pool.

Nettoyage explicite de données ZFS

La façon la plus simple de vérifier l'intégrité des données est de lancer un nettoyage explicite de toutes les données au sein du pool. Cette opération traverse toutes les données dans le pool une fois et vérifie que tous les blocs sont lisibles. Le nettoyage va aussi vite que le permettent les périphériques, mais la priorité de toute E/S reste inférieure à celle de toute opération normale. Cette opération peut affecter les performances, bien que les données du pool restent utilisables et leur réactivité quasiment la même lors du nettoyage. La commande `zpool scrub`permet de lancer un nettoyage explicite. Par exemple :

```
# zpool scrub tank
```

La commande `zpool status` ne permet pas d'afficher l'état de l'opération de nettoyage actuelle. Par exemple :

```
# 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
```

Une seule opération de nettoyage actif par pool peut se produire à la fois.

L'option `-s` permet d'interrompre une opération de nettoyage en cours. Par exemple :

```
# zpool scrub -s tank
```

Dans la plupart des cas, une opération de nettoyage pour assurer l'intégrité des données doit être menée à son terme. Vous pouvez cependant interrompre une telle opération si les performances du système sont affectées.

Un nettoyage de routine garantit des E/S continues pour l'ensemble des disques du système. Cet opération a cependant pour effet secondaire d'empêcher la gestion de l'alimentation de placer des disques inactifs en mode basse consommation. Si le système réalise en général des E/S en permanence, ou si la consommation n'est pas une préoccupation, ce problème peut être ignoré.

Pour de plus amples informations sur l'interprétation de la sortie de `zpool status`, reportez-vous à la section “Requête d'état de pool de stockage ZFS” à la page 88.

Nettoyage et réargenture de données ZFS

Lors du remplacement d'un périphérique, une opération de réargenture est amorcée pour déplacer les données des copies correctes vers le nouveau périphérique. Cette action est une forme de nettoyage de disque. Par conséquent, une seule action de ce type peut être effectuée à un moment donné dans le pool. Lorsqu'une opération de nettoyage est en cours, toute opération de réargenture suspend le nettoyage ; le nettoyage reprend une fois que la réargenture est terminée.

Pour de plus amples informations sur la réargenture, reportez-vous à la section “Affichage de l'état de réargenture” à la page 313.

Données ZFS altérées

L'altération de données se produit lorsqu'une ou plusieurs erreurs de périphériques (indiquant un ou plusieurs périphériques manquants ou endommagés) affectent un périphérique virtuel de niveau supérieur. Par exemple, la moitié d'un miroir peut subir des milliers d'erreurs sans jamais causer d'altération de données. Si une erreur se produit sur l'autre côté du miroir au même emplacement, les données sont endommagées.

L'altération de données est toujours permanente et nécessite une soin particulier lors de la réparation. Même en cas de réparation ou de remplacement des périphériques sous-jacents, les données d'origine sont irrémédiablement perdues. La plupart du temps, ce scénario requiert la restauration des données à partir de sauvegardes. Les erreurs de données sont enregistrées à mesure qu'elles sont détectées et peuvent être contrôlées à l'aide de nettoyages de pools de routine, comme expliqué dans la section suivante. Lorsqu'un bloc endommagé est supprimé, le nettoyage de disque suivant reconnaît que l'altération n'est plus présente et supprime toute trace de l'erreur dans le système.

Résolution des problèmes d'espace ZFS

Consultez les sections suivantes si vous n'êtes pas sûr de la manière dont ZFS signale le système de fichiers et la comptabilisation d'espace du pool. Consultez également la section “Comptabilisation de l'espace disque ZFS” à la page 32.

Compte-rendu d'espace de système de fichiers ZFS

Les commandes `zpool list` et `zfs list` sont plus appropriées que les commandes précédentes `df` et `du` pour déterminer l'espace disponible des pools et des systèmes de fichiers. Les anciennes commandes ne permettent pas de distinguer facilement l'espace des pools de l'espace des systèmes de fichiers. D'autre part, elles ne tiennent pas compte de l'espace utilisé par les systèmes de fichiers descendants ou les instantanés.

Par exemple, le pool root ci-après (`rpool`) utilise 5,46 Go et dispose de 68,5 Go d'espace libre.

```
# zpool list rpool
NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool   74G  5.46G  68.5G   7%  1.00x  ONLINE  -
```

Si vous comparez la comptabilisation d'espace de pool à la comptabilisation d'espace de système de fichiers en consultant la colonne `USED` de vos systèmes de fichiers individuels, vous pouvez voir que l'espace de pool qui est signalé dans `ALLOC` est globalement présent dans le total `USED` du système de fichiers. Par exemple :

```
# 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  -
```

Compte-rendu sur l'espace des pools de stockage ZFS

La valeur de taille SIZE calculée par la commande `zpool list` indique généralement la quantité d'espace disque physique dans le pool, mais elle varie selon le niveau de redondance de celui-ci. Voir les exemples ci-dessous. La commande `zfs list` liste l'espace disponible pour des systèmes de fichiers, c'est-à-dire l'espace disque moins l'espace utilisé par les métadonnées de gestion de la redondance des pools ZFS, le cas échéant.

- **Pool de stockage non redondant** : un pool est créé avec un seul disque de 136 Go ; la commande `zpool list` signale une valeur de taille `SIZE` et une valeur d'espace libre initiale `FREE` de 136 Go. L'espace disponible initial `AVAIL` indiqué par la commande `zfs list` est de 134 Go, en raison d'une petite quantité de métadonnées de gestion de pool. Par exemple :

  ```
  # 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
  ```

- **Pool de stockage mis en miroir** : quand un pool est créé avec deux disques de 136 Go , la commande `zpool list` indique une valeur de taille `SIZE` de 136 Go et une valeur d'espace libre initiale `FREE` de 136 Go. Ce compte-rendu est appelé valeur d'espace *minorée*. L'espace disponible initial `AVAIL` indiqué par la commande `zfs list` est de 134 Go, en raison d'une petite quantité de métadonnées de gestion de pool. Par exemple :

  ```
  # 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
  ```

- **Pool de stockage RAID-Z** : quand un pool `raidz2` est créé avec trois disques de 136 Go ; la commande `zpool list` signale une valeur de taille `SIZE` de 408 Go et une valeur d'espace libre initiale `FREE` de 408 Go. Ce compte-rendu est appelé valeur d'espace disque *majorée*, qui inclut l'espace nécessaire à la gestion de la redondance, par exemple les informations de parité. L'espace disponible initial `AVAIL` indiqué par la commande `zfs list` est de 133 Go, en raison de la gestion de la redondance du pool. La différence d'espace entre les sorties `zpool list` et `zfs list` pour un pool RAID-Z provient du fait que `zpool list` indique l'espace de pool majoré.

```
# 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
```

Réparation de données endommagées

Les sections suivantes décrivent comment identifier le type d'altération de données et comment réparer les données le cas échéant.

- "Identification du type d'altération de données" à la page 320
- "Réparation d'un fichier ou répertoire endommagé" à la page 321
- "Réparation de dommages présents dans l'ensemble du pool de stockage ZFS" à la page 323

ZFS utilise les données des sommes de contrôle, de redondance et d'auto-rétablissement pour minimiser le risque d'altération de données. Cependant, l'altération de données peut se produire si le pool n'est pas redondant, si l'altération s'est produite alors que le pool était endommagé ou si une série d'événements improbables a altéré plusieurs copies d'un élément de données. Quelle que soit la source, le résultat est le même : les données sont altérées et par conséquent inaccessibles. Les actions à effectuer dépendent du type de données altérées et de leurs valeurs relatives. Deux types de données peuvent être altérées :

- Métadonnées de pool : ZFS requiert une certaine quantité de données à analyser afin d'ouvrir un pool et d'accéder aux jeux de données. Si ces données sont altérées, le pool entier ou des parties de la hiérarchie du jeu de données sont indisponibles.
- Données d'objet : dans ce cas, l'altération se produit au sein d'un fichier ou périphérique spécifique. Ce problème peut rendre une partie du fichier ou répertoire inaccessible ou endommager l'objet.

Les données sont vérifiées lors des opérations normales et lors du nettoyage. Pour de plus amples informations sur la vérification de l'intégrité des données du pool, reportez-vous à la section "Contrôle de l'intégrité d'un système de fichiers ZFS" à la page 315.

Identification du type d'altération de données

Par défaut, la commande `zpool status` indique qu'une altération s'est produite, mais n'indique pas à quel endroit. Par exemple :

```
# 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
```

Toute erreur indique seulement qu'une erreur s'est produite à un moment donné. Il est possible que certaines erreurs ne soient plus présentes dans le système. Dans le cadre d'une utilisation normale, elles le sont. Certaines interruptions de service temporaires peuvent entraîner une altération de données qui est automatiquement réparée une fois l'interruption de service terminée. Un nettoyage complet du pool examine chaque bloc actif dans le pool. Ainsi, le journal d'erreur est réinitialisé à la fin de chaque nettoyage. Si vous déterminez que les erreurs ne sont plus présentes et ne souhaitez pas attendre la fin du nettoyage, la commande `zpool online` permet de réinitialiser toutes les erreurs du pool.

Si l'altération de données se produit dans des métadonnées au niveau du pool, la sortie est légèrement différente. Par exemple :

```
# 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
```

Dans le cas d'une altération au niveau du pool, ce dernier se voit attribuer l'état `FAULTED`, car le pool ne peut pas fournir le niveau de redondance requis.

Réparation d'un fichier ou répertoire endommagé

En cas d'altération d'un fichier ou d'un répertoire, le système peut tout de même continuer à fonctionner, selon le type d'altération. Tout dommage est irréversible, à moins que des copies correctes des données n'existent sur le système. Si les données sont importantes, vous devez restaurer les données affectées à partir d'une sauvegarde. Quand bien même, vous devriez pouvoir réparer les données altérées sans restaurer la totalité du pool.

En cas de dommages au sein d'un bloc de données de fichiers, le fichier peut être supprimé en toute sécurité. L'erreur est alors effacée du système. Utilisez la commande `zpool status -v` pour afficher la liste des noms de fichier contenant des erreurs persistantes. Par exemple :

```
# zpool status -v
  pool: monkey
 state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption.  Applications may be affected.
action: Restore the file in question if possible.  Otherwise restore the
        entire pool from backup.
   see: http://www.sun.com/msg/ZFS-8000-8A
 scrub: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:

        NAME        STATE     READ WRITE CKSUM
        monkey      ONLINE       8     0     0
          c1t1d0    ONLINE       2     0     0
          c2t5d0    ONLINE       6     0     0

errors: Permanent errors have been detected in the following files:

/monkey/a.txt
/monkey/bananas/b.txt
/monkey/sub/dir/d.txt
monkey/ghost/e.txt
/monkey/ghost/boo/f.txt
```

La liste des noms de fichiers comportant des erreurs persistantes peut être décrite comme suit :

- Si le chemin complet du fichier est trouvé et si le jeu de données est monté, le chemin complet du fichier s'affiche. Par exemple :

  ```
  /monkey/a.txt
  ```

- Si chemin complet du fichier est trouvé mais que le jeu de données n'est pas monté, le nom du jeu de données non précédé d'un slash (/) s'affiche, suivi du chemin du fichier au sein du jeu de données. Par exemple :

  ```
  monkey/ghost/e.txt
  ```

- Si le nombre d'objet vers un chemin de fichiers ne peut pas être converti, soit en raison d'une erreur soit parce qu'aucun chemin de fichiers réel n'est associé à l'objet, tel que c'est le cas pour `dnode_t`, alors le nom du jeu de données s'affiche, suivi du numéro de l'objet. Par exemple :

  ```
  monkey/dnode:<0x0>
  ```

- En cas d'altération d'un MOS (Meta-Object Set, jeu de méta-objet), la balise spéciale `<metadata>` s'affiche, suivie du numéro de l'objet.

Si l'altération est localisée dans les métadonnées d'un répertoire ou d'un fichier, vous devez déplacer le fichier vers un autre emplacement. Vous pouvez déplacer en toute sécurité les fichiers ou les répertoires vers un autre emplacement. Cela permet de restaurer l'objet d'origine à son emplacement.

Réparation de données endommagées avec plusieurs références de blocs

Si un système de fichiers endommagé contient des données endommagées avec plusieurs références de blocs tels que les instantanés, la commande `zpool status -v` ne peut pas afficher les chemins de **toutes** les données endommagées. La génération de rapports `zpool status` actuelle de données endommagées est limitée par le niveau d'altération de métadonnées et par la réutilisation d'un bloc après l'exécution de la commande `zpool status`. Les blocs dédupliqués rendent la génération de rapports sur les données endommagées encore plus complexe.

Si vous avez des données endommagées et que la commande `zpool status -v` signale que les données d'instantané sont affectées, pensez à exécuter la commande suivante afin d'identifier les autres chemins endommagés :

Réparation de dommages présents dans l'ensemble du pool de stockage ZFS

Si des dommages sont présents dans les métadonnées du pool et que cela empêche l'ouverture ou l'importation du pool, vous pouvez utiliser les options suivantes :

- Tentez de récupérer le pool à l'aide de la commande `zpool clear -F` ou `zpool import -F`. Ces commandes tentent d'annuler (roll back) les dernières transactions restantes du pool pour qu'elles reviennent à un fonctionnement normal. Vous pouvez utiliser la commande `zpool status` pour vérifier le pool endommagé et les mesures de récupération recommandées. Par exemple :

```
# 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
```

Le processus de récupération comme décrit dans la sortie ci-dessus consiste à utiliser la commande suivante :

```
# zpool clear -F tpool
```

Si vous tentez d'importer un pool de stockage endommagé, des messages semblables aux messages suivants s'affichent :

```
# 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.
```

Le processus de récupération comme décrit dans la sortie ci-dessus consiste à utiliser la commande suivante :

```
# zpool import -F tpool
Pool tpool returned to its state as of Wed Jul 14 11:44:10 2010.
Discarded approximately 5 seconds of transactions
```

Si le pool endommagé se trouve dans le fichier `zpool.cache`, le problème est détecté lors de l'initialisation du système. Le pool endommagé est consigné dans la commande `zpool status`. Si le pool ne se trouve pas dans le fichier `zpool.cache`, il n'est pas importé ou ouvert et des messages indiquant que le pool est endommagé s'affichent lorsque vous tentez de l'importer.

- Vous pouvez importer un pool endommagé en mode lecture seule. Cette méthode permet d'importer le pool, ce qui vous permet d'accéder aux données. Par exemple :

  ```
  # zpool import -o readonly=on tpool
  ```

  Pour plus d'informations sur l'importation d'un pool en lecture seule, reportez-vous à la section "Importation d'un pool en mode lecture seule" à la page 106.

- Vous pouvez importer un pool avec un périphérique de journalisation manquant à l'aide de la commande `zpool import -m`. Pour plus d'informations, reportez-vous à la section "Importation d'un pool avec un périphérique de journalisation manquant" à la page 105.

- Si le pool ne peut pas être récupéré par le biais de l'une des méthodes de récupération de pool, vous devez restaurer le pool et l'ensemble de ses données à partir d'une copie de sauvegarde. Le mécanisme utilisé varie énormément selon la configuration du pool et la stratégie de sauvegarde. Tout d'abord, enregistrez la configuration telle qu'elle s'affiche dans la commande `zpool status` pour pouvoir la recréer après la destruction du pool. Ensuite, détruisez le pool à l'aide de la commande `zpool destroy -f`.

  Conservez également un fichier décrivant la disposition des jeux de données et les diverses propriétés définies localement dans un emplacement sûr, car ces informations deviennent inaccessibles lorsque le pool est lui-même inaccessible. Avec la configuration du pool et la disposition des jeux de données, vous pouvez reconstruire la configuration complète après destruction du pool. Les données peuvent ensuite être renseignées par la stratégie de sauvegarde ou de restauration de votre choix.

Réparation d'une configuration ZFS endommagée

ZFS gère un cache de pools actifs et les configurations correspondantes dans le système de fichiers root. Si ce fichier de cache est endommagé ou n'est plus synchronisé avec les informations de configuration stockées dans le disque, l'ouverture du pool n'est plus possible. Le système de fichiers ZFS tente d'éviter cette situation, même si des altérations arbitraires peuvent toujours survenir en raison des caractéristiques du système de fichiers sous-jacent et du stockage. En général, cette situation est due à la disparition d'un pool du système alors qu'il devrait être disponible. Parfois, elle correspond à une configuration partielle, dans laquelle il manque un nombre inconnu de périphériques virtuels de niveau supérieur. Quel que soit le cas, la configuration peut être récupérée en exportant le pool (s'il est visible à tous) et en le réimportant.

Pour de plus amples informations sur l'importation et l'exportation de pools, reportez-vous à la section "Migration de pools de stockage ZFS" à la page 100.

Réparation d'un système impossible à réinitialiser

ZFS a été conçu pour être robuste et stable malgré les erreurs. Cependant, les bogues de logiciels ou certains problèmes inattendus peuvent entraîner la panique du système lors de l'accès à un pool. Dans le cadre du processus d'initialisation, chaque pool doit être ouvert. En raison de ces défaillances, le système effectue des réinitialisations en boucle. Pour pouvoir reprendre les opérations dans cette situation, vous devez indiquer à ZFS de ne pas rechercher de pool au démarrage.

ZFS conserve un cache interne de pools disponibles et de leurs configurations dans `/etc/zfs/zpool.cache`. L'emplacement et le contenu de ce fichier sont privés et sujets à modification. Si le système devient impossible à initialiser, redémarrez au jalon `none` à l'aide de l'option d'initialisation `-m milestone=none`. Une fois le système rétabli, remontez le système de fichiers root en tant que système accessible en écriture, puis renommez ou placez le fichier `/etc/zfs/zpool.cache` à un autre emplacement. En raison de ces actions, ZFS oublie l'existence de pools dans le système, ce qui l'empêche d'accéder au pool défectueux à l'origine du problème. Vous pouvez ensuite passer à un état normal de système en exécutant la commande `svcadm milestone all`. Vous pouvez utiliser un processus similaire lors de l'initialisation à partir d'un root de remplacement pour effectuer des réparations.

Une fois le système démarré, vous pouvez tenter d'importer le pool à l'aide de la commande `zpool import`. Cependant, dans ce cas, l'erreur qui s'est produite lors de l'initialisation risque de se reproduire car la commande utilise le même mécanisme d'accès aux pools. Si le système contient plusieurs pools, procédez comme suit :

- Renommez ou déplacez le fichier `zpool.cache` vers un autre emplacement comme décrit dans le paragraphe ci-dessus.

- Utilisez la commande `fmdump -eV` pour afficher les pools présentant des erreurs fatales et déterminer ainsi quel pool pose des problèmes.
- Importez les pools un à un en ignorant ceux qui posent problème, comme décrit dans la sortie de la commande `fmdump`.

◆ ◆ ◆ CHAPITRE 11

Pratiques recommandées pour Oracle Solaris ZFS

Ce chapitre décrit les pratiques recommandées pour la création, la surveillance et la gestion de pools de stockage ZFS et de systèmes de fichiers.

Ce chapitre contient les sections suivantes :

- "Pratiques recommandées pour les pools de stockage" à la page 327
- "Pratiques recommandées pour les systèmes de fichiers" à la page 335

Pratiques recommandées pour les pools de stockage

Les sections suivantes décrivent les pratiques recommandées pour la création et la surveillance de pools de stockage ZFS. Pour plus d'informations sur le dépannage des problèmes de pools de stockage, reportez-vous au Chapitre 10, "Dépannage d'Oracle Solaris ZFS et récupération de pool".

Pratiques recommandées générales

- Maintenez votre système à jour grâce aux dernières versions et aux correctifs Solaris.
- Confirmez que votre contrôleur honore les commandes de purge de cache pour être sûr que vos données ont été écrites de manière sécurisée, ce qui permet de modifier les périphériques du pool ou de séparer un pool de stockage mis en miroir. Ce n'est généralement pas un problème sur le matériel Oracle/Sun, mais il est bon de confirmer que les paramètres de purge du cache de votre matériel sont activés.
- Evaluez la mémoire requise en fonction de la charge de travail actuelle du système.
  - Avec une application dont l'encombrement mémoire est connu, telle qu'une application de base de données par exemple, vous pouvez limiter la taille de l'ARC de manière à ce que l'application n'ait pas besoin de récupérer la mémoire nécessaire à partir du cache ZFS.

- Tenez compte de la mémoire requise pour la suppression des doublons.
- Identifiez l'utilisation de la mémoire par ZFS à l'aide de la commande suivante :

```
# 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
```

- Envisagez l'utilisation de mémoire ECC pour prévenir l'altération de mémoire. L'altération de mémoire silencieuse peut endommager vos données.

- Effectuez des sauvegardes régulières : bien qu'un pool créé avec redondance ZFS permette de réduire le temps d'inactivité dû à des pannes matérielles, il n'est pas à l'abri des pannes matérielles, des pannes d'alimentation ou des déconnexions de câbles. Veillez à sauvegarder régulièrement vos données. Toutes les données importantes doivent être sauvegardées. Il existe différentes méthodes de copie des données :
  - Prise quotidienne ou à intervalles réguliers d'instantanés ZFS.
  - Sauvegardes hebdomadaires des données du pool ZFS. Vous pouvez utiliser la commande `zpool split` pour créer une copie exacte du pool de stockage ZFS mis en miroir.
  - Sauvegardes mensuelles à l'aide d'un produit de sauvegarde mis en oeuvre à l'échelle de l'entreprise.
- RAID matériel
  - Envisagez l'utilisation du mode JBOD pour les baies de stockage plutôt que des baies RAID matérielles, afin que ZFS puisse gérer le stockage et la redondance.
  - Utilisez la redondance matérielle RAID et/ou la redondance ZFS.
  - L'utilisation de la redondance ZFS présente de nombreux avantages : pour les environnements de production, configurez ZFS de manière à lui permettre de réparer les incohérences de données. Utilisez la redondance ZFS, telle que RAID-Z, RAID-Z-2, RAID-Z-3, la mise en miroir, quel que soit le niveau RAID mis en oeuvre sur le périphérique de stockage sous-jacent. Avec une telle redondance, ZFS est en mesure de détecter et de réparer les défaillances du périphérique de stockage sous-jacent ou des connexions à l'hôte de celui-ci.

Reportez-vous également à la section “Pratiques recommandées pour la création de pool sur une baie de stockage en réseau ou locale” à la page 331.

- Les vidages sur incident consomment davantage d'espace disque, généralement entre 1/2 et 3/4 de la taille de la mémoire physique.

Pratiques de création de pools de stockage ZFS

Les sections suivantes présentent des pratiques recommandées générales et plus spécifiques pour les pools de stockage

Pratiques recommandées générales pour les pools de stockage

- Utilisez des disques entiers pour activer la mise en cache et faciliter la maintenance. La création de pools sur des tranches complique la gestion et la récupération des disques.
- Utilisez la redondance ZFS pour permettre à ZFS de réparer les incohérences de données.
  - Le message suivant s'affiche lorsqu'un pool non redondant est créé :

    ```
    # zpool create tank c4t1d0 c4t3d0
    'tank' successfully created, but with no redundancy; failure
    of one device will cause loss of the pool
    ```
  - Pour des pools mis en miroir, utilisez des paires de disques mis en miroir
  - Pour des pools RAID-Z, regroupez 3 à 9 disques par VDEV
  - Ne mélangez pas les composants RAID-Z et mis en miroir dans un même pool. Ces pools sont plus difficiles à gérer et les performances peuvent en souffrir.
- Utilisez des disques hot spare pour réduire le temps d'inactivité dû aux pannes matérielles.
- Utilisez des disques de taille similaire afin que de répartir les E/S de façon équilibrée entre les périphériques.
  - Des LUN de petite taille peuvent être étendus en LUN de grande taille
  - N'étendez pas des LUN de façon excessive, comme par exemple de 128 Mo à 2 To, de manière à conserver des tailles de metaslabs optimales
- Envisagez la création d'un petit pool root et de pools de données plus volumineux pour assurer une récupération plus rapide du système
- La taille de pool minimale recommandée est de 8 Go. Bien que la taille de pool minimale est de 64 Mo, toute taille inférieure à 8 Go rend l'allocation et la demande d'espace de pool libre plus difficiles.
- La taille de pool maximale recommandée doit être adaptée à votre charge de travail ou à la taille de vos données. N'essayez pas de stocker plus de données que vous ne pouvez en sauvegarder régulièrement. Dans le cas contraire, vos données sont vulnérables en cas d'événement imprévu.

Pratiques recommandées pour la création de pool root

- Créez des pools root comportant des tranches à l'aide de l'identificateur `s*`. N'utilisez pas l'identificateur `p*`. Le pool root ZFS d'un système est généralement créé au moment de l'installation du système. Si vous êtes en train de créer un second pool root ou de recréer un pool root, utilisez une syntaxe semblable à la suivante :

  ```
  # zpool create rpool c0t1d0s0
  ```

  Sinon, créez un pool root mis en miroir. Par exemple :

  ```
  # zpool create rpool mirror c0t1d0s0 c0t2d0s0
  ```

  - Le pool root doit être créé sous la forme d'une configuration en miroir ou d'une configuration à disque unique. Les configurations RAID-Z ou entrelacées ne sont pas prises en charge. Vous ne pouvez pas ajouter d'autres disques mis en miroir pour créer plusieurs périphériques virtuels de niveau supérieur à l'aide de la commande `zpool add`. Toutefois, vous pouvez étendre un périphérique virtuel mis en miroir à l'aide de la commande `zpool attach`.
  - Un pool root ne peut pas avoir de périphérique de journalisation distinct.
  - Les propriétés d'un pool peuvent être définies lors d'une installation AI, mais l'algorithme de compression `gzip` n'est pas pris en charge sur les pools root.
  - Ne renommez pas le pool root une fois qu'il a été créé par une installation initiale. Si vous renommez le pool root, cela peut empêcher l'initialisation du système.
  - Ne créez pas de pool root sur une clé USB pour un système de production, car les disques de pool roots sont vitaux pour un fonctionnement continu, en particulier dans un environnement professionnel. Envisagez d'utiliser les disques internes d'un système pour le pool root, ou au moins d'utiliser des disques de la même qualité que celle que vous utiliseriez pour vos données non-roots. De plus, un clé USB peut s'avérer trop petite pour gérer une taille de fichier de vidage équivalente à la moitié de la taille de la mémoire physique.

Pratiques recommandées pour la création de pools non-root

- Créez des pools non-root avec des disques entiers à l'aide de l'identificateur `d*`. N'utilisez pas l'identificateur `p*`.
  - ZFS fonctionne mieux sans logiciel de gestion de volumes supplémentaire.
  - Pour de meilleures performances, utilisez des disques individuels ou, tout au moins, des LUN constitués d'un petit nombre de disques. En offrant à ZFS un meilleur aperçu de la configuration des LUN, vous lui permettez de prendre de meilleures décisions de planification d'E/S.
  - Créez des configurations de pools redondants dans plusieurs contrôleurs afin de réduire le temps d'inactivité dû à une panne de contrôleur.
  - **Pools de stockage mis en miroir** : consomment davantage d'espace disque mais présentent de meilleures performances pour les petites lectures aléatoires.

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

- **Pools de stockage RAID-Z** : ces pools peuvent être créés avec 3 stratégies de parité, d'une parité égale à 1 (`raidz`), 2 (`raidz2`) ou 3 (`raidz3`). Une configuration RAID-Z optimise l'espace disque et fournit généralement des performances satisfaisantes lorsque les données sont écrites et lues en gros blocs (128 Ko ou plus).
  - Prenons l'exemple d'une configuration RAID-Z à parité simple (`raidz`) avec 2 périphériques virtuels à 3 disques (2+1) chacun.

    ```
    # zpool create rzpool raidz1 c1t0d0 c2t0d0 c3t0d0 raidz1 c1t1d0 c2t1d0 c3t1d0
    ```
  - Une configuration RAIDZ-2 améliore la disponibilité des données et offre les mêmes performances qu'une configuration RAID-Z. En outre, sa valeur de temps moyen entre pertes de données MTTDL (Mean Time To Data Loss) est nettement meilleure que celle d'une configuration RAID-Z ou de miroirs bidirectionnels. Créez une configuration RAID-Z à double parité RAID-Z (`raidz2`) à 6 disques (4+2).

    ```
    # zpool create rzpool raidz2 c0t1d0 c1t1d0 c4t1d0 c5t1d0 c6t1d0 c7t1d0
    raidz2 c0t2d0 c1t2d0 c4t2d0 c5t2d0 c6t2d0 c7t2d
    ```
  - La configuration RAIDZ-3 optimise l'espace disque et offre une excellente disponibilité car elle peut résister à 3 pannes de disque. Créez une configuration RAID-Z à triple parité (`raidz3`) à 9 disques (6+3).

    ```
    # zpool create rzpool raidz3 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0
    c5t0d0 c6t0d0 c7t0d0 c8t0d0
    ```

Pratiques recommandées pour la création de pool sur une baie de stockage en réseau ou locale

Tenez compte des pratiques recommandées pour la création d'un pool root ZFS sur une baie de stockage connectée localement ou à distance.

- Si vous créez un pool sur des périphériques SAN et que la connexion réseau est lente, les périphériques du pool peuvent devenir `UNAVAIL` pendant un certain temps. Vous devez évaluer si la connexion réseau est adéquate pour fournir vos données de manière continue. En outre, considérez, si vous utilisez des périphériques SAN pour votre pool root, qu'ils peuvent ne pas être disponibles dès l'initialisation du système et que les périphériques du pool root peuvent également être `UNAVAIL`.
- Confirmez avec votre vendeur de baies que la baie de stockage ne vide pas son cache après une demande de mise en cache des enregistrements de vidage par ZFS.
- Utilisez des disques entiers, plutôt que des tranches de disque, comme périphériques de pool de stockage afin qu'Oracle Solaris ZFS active les caches des petits disques locaux, qui sont vidés à des moments appropriés.
- Pour de meilleures performances, créez un LUN pour chaque disque physique de la baie. Si vous utilisez un LUN unique et volumineux, ZFS risque de mettre trop peu d'opérations d'E/S de lecture en attente pour effectivement assurer des performances optimales de stockage. A l'inverse, l'utilisation de nombreux petits LUN peut entraîner l'inondation du stockage avec un grand nombre d'opérations d'E/S de lecture en attente.

- Une baie de stockage qui utilise un logiciel d'approvisionnement dynamique (fin) pour implémenter une allocation d'espace virtuel n'est pas recommandée pour Oracle Solaris ZFS. Quand Oracle Solaris ZFS écrit les données modifiées dans l'espace libre, il écrit sur le LUN entier. Le processus d'écriture d'Oracle Solaris ZFS alloue tout l'espace virtuel depuis le point de vue de la baie de stockage, ce qui annule l'avantage de l'approvisionnement dynamique.

  Considérez qu'un logiciel d'approvisionnement dynamique peut s'avérer inutile lors de l'utilisation de ZFS.

  - Vous pouvez étendre un LUN dans un pool de stockage ZFS existant et il utilisera le nouvel espace.
  - Un comportement similaire fonctionne quand un LUN plus petit est remplacé par un LUN plus gros.
  - Si vous évaluez les besoins de stockage pour votre pool et créez le pool avec des LUN plus petits qui correspondent aux besoins de stockage, vous pouvez alors toujours étendre les LUN si vous avez besoin de plus d'espace.
- Si la baie peut présenter des périphériques individuels (mode JBOD), envisagez de créer des pools de stockage ZFS redondants (en miroir ou RAID-Z) sur ce type de baie afin que ZFS puisse signaler et corriger les incohérences de données.

Pratiques recommandées pour la création de pools pour une base de données Oracle

Tenez compte des pratiques recommandées pour la création de pools de stockage suivantes lorsque vous créez une base de données Oracle.

- Utilisez un pool mis en miroir ou un RAID matériel pour plusieurs pools
- Les pools RAID-Z ne sont généralement pas recommandés pour les charges de travail en lecture aléatoire
- Créez un petit pool distinct avec un périphérique de journalisation distinct pour les fichiers de journalisation de la base de données
- Créez un petit pool distinct pour le journal d'archivage

Pour plus d'informations, consultez le livre blanc suivant :

`http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle`

Utilisation de pools de stockage ZFS dans VirtualBox

- Virtual Box est configuré pour ignorer des commandes de vidage de cache à partir du stockage sous-jacent par défaut. Cela signifie que dans le cas d'un blocage système ou d'une défaillance matérielle, les données peuvent être perdues.
- Activez le vidage du cache dans Virtual Box à l'aide de la commande suivante :

```
VBoxManage setextradata <VM_NAME> "VBoxInternal/Devices/<type>/0/LUN#<n>/Config/IgnoreFlush" 0
```

- <VM_NAME> est le nom de la machine virtuelle.
- <type> est le type de contrôleur, `piix3ide`, si vous utilisez le contrôleur virtuel IDE habituel, ou `ahci`, si vous utilisez un contrôleur SATA
- <n> est le numéro de disque.

Pratiques recommandées pour l'optimisation des performances des pools de stockage

- N'utilisez pas plus de 80 % de la capacité d'un pool pour des performances optimales.
- Les pools mis en miroir sont à préférer aux pools RAID-Z pour des charges de travail en lecture/écriture aléatoires
- Périphériques de journalisation distincts
  - Recommandés pour améliorer les performances d'écriture synchrone
  - Avec une charge d'écriture synchrone élevée, empêche la fragmentation de l'écriture de nombreux blocs de journal dans le pool principal
- Des périphériques de cache distincts sont recommandés pour améliorer les performances de lecture
- Nettoyage/réargenture : un très grand pool RAID-Z comportant un grand nombre de périphériques nécessite des temps de nettoyage et de réargenture plus long
- Si les performances du pool sont ralenties : utilisez la commande `zpool status` pour éliminer les problèmes matériels à l'origine des problèmes de performances du pool. Si la commande `zpool status` ne fait apparaître aucun problème, utilisez la commande `fmdump` pour afficher les pannes matérielles ou utilisez la commande `fmdump -eV` pour repérer les éventuelles défaillances matérielles qui n'ont pas encore été signalées en tant qu'erreur.

Pratiques recommandées pour la maintenance et la gestion d'un pool de stockage ZFS

- Assurez-vous que la capacité d'un pool est inférieure à 80 %, pour obtenir de meilleures performances.

  Les performances d'un pool peuvent se dégrader lorsque le pool est très plein et que les systèmes de fichiers sont fréquemment mis à jour, comme c'est le cas par exemple pour un serveur de courrier très actif. Des pools pleins peuvent entraîner une baisse des performances, mais aucun autre problème. Si la charge de travail principale consiste en des fichiers immuables, maintenez un taux d'utilisation du pool de 95 à 96 %. Même avec un contenu essentiellement statique et un taux d'utilisation de 95 à 96 %, les performances d'écriture, de lecture et de réargenture risquent de se dégrader.

    - Surveillez l'espace des pools et des systèmes de fichiers pour vous assurer qu'il n'est pas entièrement utilisé.
    - Vous pouvez envisager d'utiliser des quotas et des réservations ZFS afin d'être sûr que l'espace du système de fichiers ne dépasse pas 80 % de la capacité du pool.
- Surveillez la santé du pool
    - Surveillez les pools redondants sur une base hebdomadaire à l'aide de `zpool status` et `fmdump`
    - Surveillez les pools non redondants toutes les deux semaines à l'aide de `zpool status` et `fmdump`
- Exécutez régulièrement `zpool scrub` pour repérer les problèmes d'intégrité des données.
    - Si vous utilisez des lecteurs de qualité grand public, envisagez de planifier un nettoyage hebdomadaire.
    - Si vous utilisez des lecteurs de qualité professionnelle, envisagez de planifier un nettoyage hebdomadaire.
    - Vous devez également exécuter un nettoyage avant de remplacer des périphériques ou de réduire temporairement la redondance d'un pool, afin d'assurer que tous les périphériques sont alors opérationnels.
- Surveillance des défaillances de pools ou de périphériques : utilisez `zpool status` comme décrit ci-dessous. Utilisez également les commandes `fmdump` ou `fmdump -eV` pour vérifier l'absence de défauts et d'erreurs au niveau des périphériques.
    - Surveillez la santé des pools redondants sur une base hebdomadaire à l'aide de `zpool status` et `fmdump`
    - Surveillez la santé des pools non redondants toutes les deux semaines à l'aide de `zpool status` et `fmdump`
- Le périphérique de pool est `UNAVAIL` ou `OFFLINE` : si un périphérique de pool n'est pas disponible, vérifiez si le périphérique est répertorié dans la sortie de la commande `format`. Si le périphérique n'apparaît pas dans la sortie de `format`, il n'est pas visible sur ZFS.

  L'état de périphérique de pool `UNAVAIL` ou `OFFLINE` signifie généralement que le périphérique est en panne, qu'un câble est déconnecté ou qu'un autre problème matériel, tel qu'un câble ou un contrôleur défectueux, a rendu inaccessible le périphérique.
- Surveillez l'espace du pool de stockage : utilisez les commandes `zpool list` et `zfs list` pour déterminer la quantité d'espace disque utilisée par les données des systèmes de fichiers. Les instantanés ZFS peuvent consommer de l'espace disque et, lorsqu'ils ne sont pas répertoriés par la commande `zfs list`, peuvent consommer de l'espace disque de manière silencieuse. Utilisez la commande d'instantané `zfs list -t` pour identifier l'espace disque consommé par des instantanés.

Pratiques recommandées pour les systèmes de fichiers

Les sections suivantes décrivent les pratiques recommandées pour les systèmes de fichiers.

Pratiques recommandées pour la création de systèmes de fichiers

Les sections suivantes décrivent les pratiques recommandées pour la création de systèmes de fichiers ZFS.

- Créez un système de fichiers par utilisateur pour les répertoires personnels
- Envisagez d'utiliser des quotas et des réservations de systèmes de fichiers pour gérer et réserver de l'espace disque pour les systèmes de fichiers importants.
- Envisagez de définir des quotas par utilisateur ou par groupe pour gérer l'espace disque dans un environnement comptant de nombreux utilisateurs.
- Utilisez l'héritage des propriétés ZFS pour appliquer des propriétés à un grand nombre de systèmes de fichiers descendants.

Pratiques recommandées pour la création de systèmes de fichiers pour une base de données Oracle

Tenez compte des pratiques recommandées pour la création de systèmes de fichiers suivantes lorsque vous créez une base de données Oracle.

- Faites correspondre la propriété `recordsize` ZFS à la taille `db_block_size` Oracle.
- Créez des systèmes de fichiers pour la table de base données et l'index dans le pool de base de données principal, en utilisant une taille `recordsize` de 8 Ko et la valeur `primarycache` par défaut.
- Créez des systèmes de fichiers pour les données temporaires et l'espace de la table d'annulation dans le pool de base de données principal, en utilisant les valeurs `recordsize` et `primarycache` par défaut.
- Créez un système de fichiers pour le journal d'archivage dans le pool d'archivage, en activant la compression et la valeur `recordsize` par défaut et en définissant `primarycache` sur `metadata`.

Pour plus d'informations, consultez le livre blanc suivant :

`http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle`

Pratiques recommandées pour la surveillance de systèmes de fichiers ZFS

Il est recommandé de surveiller les systèmes de fichiers ZFS pour s'assurer qu'ils sont disponibles et pour identifier les problèmes de consommation d'espace.

- Une fois par semaine, contrôlez l'espace disponible des systèmes de fichiers à l'aide des commandes `zpool list` et `zfs list` au lieu des commandes `du` et `df` car les anciennes commandes ne tiennent pas compte de l'espace consommé par les systèmes de fichiers descendants ou les instantanés.

  Pour plus d'informations, reportez-vous à la section "Résolution des problèmes d'espace ZFS" à la page 318.
- Affichez la consommation d'espace des systèmes de fichiers à l'aide de la commande `zfs list -o space`.
- L'espace des systèmes de fichiers peut être consommé par des instantanés à votre insu. Vous pouvez afficher toutes les informations sur les systèmes de fichiers en respectant la syntaxe suivante :

  ```
  # zfs list -t all
  ```
- Un système de fichiers `/var` distinct est automatiquement créé lorsqu'un système est installé, mais il est recommandé de définir un quota et une réservation sur ce système de fichiers pour s'assurer qu'il ne consomme pas d'espace de pool root à votre insu.
- En outre, vous pouvez utiliser la commande `fsstat` pour afficher les activités de traitement de fichiers des systèmes de fichiers ZFS. Les activités peuvent être consignées par point de montage ou par type de système de fichiers. L'exemple suivant illustre les activités générales de système de fichiers ZFS :

  ```
  # 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 /
  ```
- Sauvegardes
  - Conservez des instantanés des systèmes de fichiers
  - Envisagez d'effectuer des sauvegardes hebdomadaires et mensuelles à l'aide de logiciels mis en oeuvre à l'échelle de l'entreprise
  - Stockez des instantanés des pools root sur un système distant pour la récupération à chaud

◆ ◆ ◆ ANNEXE A

Descriptions des versions d'Oracle Solaris ZFS

Cette annexe décrit les versions ZFS disponibles, les fonctionnalités de chacune d'entre elles et le SE Solaris correspondant.

L'annexe contient les sections suivantes :

- "Présentation des versions ZFS" à la page 337
- "Versions de pool ZFS" à la page 337
- "Versions du système de fichiers ZFS" à la page 339

Présentation des versions ZFS

Les nouvelles fonctions de pool et de système de fichiers ZFS ont été introduites dans certaines versions du système de fichiers ZFS disponibles avec Solaris. Vous pouvez utiliser la commande `zpool upgrade` ou `zfs upgrade` pour déterminer si la version d'un pool ou un système de fichiers est antérieure à la version Solaris en cours d'exécution. Vous pouvez également utiliser ces commandes pour mettre à niveau la version de votre pool et de votre système de fichiers.

Pour plus d'informations sur l'utilisation des commandes `zpool upgrade` et `zfs upgrade`, reportez-vous aux sections "Mise à niveau des systèmes de fichiers ZFS" à la page 218 et "Mise à niveau de pools de stockage ZFS" à la page 109.

Versions de pool ZFS

Le tableau suivant présente une liste des versions de pool ZFS disponibles dans la version d'Oracle Solaris.

Version	Solaris 10	Description
1	Solaris 10 6/06	Version ZFS initiale

Version	Solaris 10	Description
2	Solaris 10 11/06	Blocs ditto (métadonnées répliquées)
3	Solaris 10 11/06	Disques hot spare et RAID-Z à double parité
4	Solaris 10 8/07	`zpool history`
5	Solaris 10 10/08	Algorithme de compression `gzip`
6	Solaris 10 10/08	Propriété de pool `bootfs`
7	Solaris 10 10/08	Périphérique de journalisation de tentatives distincts
8	Solaris 10 10/08	Administration déléguée
9	Solaris 10 10/08	Propriétés `refquota` et `refreservation`
10	Solaris 10 5/09	Périphériques de cache
11	Solaris 10 10/09	Performances de nettoyage améliorées
12	Solaris 10 10/09	Propriétés d'instantané
13	Solaris 10 10/09	Propriété `snapused`
14	Solaris 10 10/09	Propriété `aclinherit` `passthrough-x`
15	Solaris 10 10/09	Comptabilisation des espaces de groupe et d'utilisateur
16	Solaris 10 9/10	Prise en charge de la propriété `stmf`
17	Solaris 10 9/10	RAID-Z triple parité
18	Solaris 10 9/10	Instantanés conservés par l'utilisateur
19	Solaris 10 9/10	Suppression d'un périphérique de journalisation
20	Solaris 10 9/10	`Compression` `using` `zle` (encodage de chaîne vide)
21	Solaris 10 9/10	Réservé
22	Solaris 10 9/10	Propriétés reçues
23	Solaris 10 8/11	Slim ZIL
24	Solaris 10 8/11	Attributs système
25	Solaris 10 8/11	Statistiques de nettoyage améliorées
26	Solaris 10 8/11	Performances de suppression des instantanés améliorées
27	Solaris 10 8/11	Performances de création d'instantanés améliorées
28	Solaris 10 8/11	Remplacements de vdev multiples
29	Solaris 10 8/11	Programme d'allocation hybride de miroir/RAID-Z

Version	Solaris 10	Description
30	Solaris 10 1/13	Réservé
31	Solaris 10 1/13	Performances `zfs list` améliorées
32	Solaris 10 1/13	Taille de bloc d'1 Mo

Versions du système de fichiers ZFS

Le tableau suivant répertorie les versions du système de fichiers ZFS disponibles dans la version d'Oracle Solaris. N'oubliez pas qu'une fonction disponible dans certaines versions du système de fichiers nécessite une version de pool spécifique.

Version	Solaris 10	Description
1	Solaris 10 6/06	Version initiale du système de fichiers ZFS
2	Solaris 10 10/08	Entrées de répertoire améliorées
3	Solaris 10 10/08	Insensibilité à la casse et identificateur unique de système de fichiers (FUID)
4	Solaris 10 10/09	Propriétés `userquota` et `groupquota`
5	Solaris 10 8/11	Attributs système

Index

A

Accès
  Instantané ZFS
    Exemple, 223
ACL
  ACL dans un fichier ZFS
    Description détaillée, 249, 250
  `aclinherit`, propriété, 247
  Configuration d'ACL dans un fichier ZFS (mode compact)
    Description, 261
    Exemple, 262
  Configuration d'héritage d'ACL dans un fichier ZFS (mode détaillé)
    Exemple, 255
  Configuration de fichiers ZFS
    Description, 248
  Définition des ACL sur un fichier ZFS (mode détaillé)
    Description, 251
  Description, 241
  Description de format, 243
  Différences avec les ACL POSIX-draft, 242
  Héritage d'ACL, 246
  Indicateur d'héritage d'ACL, 246
  Modification d'ACL triviale dans un fichier ZFS (mode détaillé)
    (Exemple), 252
  Privilèges d'accès, 244
  Propriété d'ACL, 247
  Restauration d'une ACL triviale sur un fichier ZFS (mode détaillé)
ACL, Restauration d'une ACL triviale sur un fichier ZFS (mode détaillé) *(Suite)*
    Exemple, 254
  Type d'entrée, 244
ACL, mode de propriété, `aclinherit`, 184
ACL NFSv4
  Description de format, 243
  Différences avec les ACL POSIX-draft, 242
  Héritage d'ACL, 246
  Indicateur d'héritage d'ACL, 246
  Modèle
    Description, 241
  Propriété d'ACL, 247
ACL POSIX-draft, Description, 242
ACL Solaris
  Description de format, 243
  Différences avec les ACL POSIX-draft, 242
  Héritage d'ACL, 246
  Indicateur d'héritage d'ACL, 246
  Nouveau modèle
    Description, 241
  Propriété d'ACL, 247
`aclinherit`, propriété, 247
Administration déléguée , présentation, 267
Administration déléguée de ZFS, présentation, 267
Administration simplifiée, Description, 28
Affichage
  Autorisations déléguées (exemple), 276
  Etat de maintenance d'un pool de stockage ZFS
    Exemple, 96
  Etat détaillé du fonctionnement du pool de stockage ZFS

Affichage, Etat détaillé du fonctionnement du pool de stockage ZFS *(Suite)*
Exemple, 97
Etat fonctionnel d'un pool de stockage
Description, 95
Rapport syslog de messages d'erreur ZFS
Description, 296
Statistiques d'E/S à l'échelle du pool de stockage ZFS
Exemple, 93
Statistiques d'E/S de pool de stockage vdev ZFS
Exemple, 94
Statistiques d'E/S de pools de stockage ZFS
Description, 92
Ajout
Disques, configuration RAID-Z (exemple), 66
Périphérique à un pool de stockage ZFS (`zpool add`)
Exemple, 64
Périphérique de journalisation mis en miroir (exemple), 66
Périphériques de cache (exemple), 68
Système de fichiers ZFS à une zone non globale
Exemple, 286
Volume ZFS à une zone non globale (exemple), 287
Ajustement, Tailles des périphériques de swap et de vidage, 159
`allocated` (propriété), description, 86
`altroot`, propriété (description), 86
Annulation du partage
Système de fichiers ZFS
exemple, 211
`atime`, Propriété, Description, 184
`autoreplace` (propriété), description, 86
Autorétablissement, Description, 51
`available` (propriété), Description, 184

B

Bloc d'initialisation, Installation avec `installboot` et `installgrub`, 163
`bootfs`, propriété, description, 86

C

`cachefile` (propriété), description, 87
`canmount`, propriété, Description, 185
`canmount` (propriété), Description détaillée, 195
`checksum`, propriété, Description, 185
Clone
Création, exemple, 227
Définition, 29
Destruction, exemple, 228
Fonction, 227
Comportement d'espace saturé, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 34
Composant ZFS, Convention d'attribution de nom, 31
Composants, Pool de stockage ZFS, 45
`compression`, propriété, Description, 185
`compressratio`, propriété, Description, 185
Comptabilisation de l'espace ZFS, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 33
Configuration
ACL dans un fichier ZFS
Description, 248
ACL dans un fichier ZFS (mode compact)
Description, 261
Exemple, 262
Héritage d'ACL dans un fichier ZFS (mode détaillé)
Exemple, 255
Configuration en miroir
Description, 49
Fonction de redondance, 49
Vue conceptuelle, 49
Configuration RAID-Z
Double parité, description, 49
Exemple, 54
Fonction de redondance, 49
Parité simple, description, 49
Vue conceptuelle, 49
Configuration RAID-Z, ajout de disques, Exemple, 66
Configuration requise, Installation et Oracle Solaris Live Upgrade, 113
Connexion
Périphérique, à un pool de stockage ZFS (`zpool attach`)

Connexion, Périphérique, à un pool de stockage ZFS (`zpool attach`) *(Suite)*
Exemple, 69
Contrôle
Intégrité des données ZFS, 316
Validation des données (nettoyage), 316
Convention d'attribution de nom, Composant ZFS, 31
`copies`, propriété, Description, 186
Création
Clone ZFS, exemple, 227
Hiérarchie d'un système de fichiers ZFS, 41
création
Instantané ZFS
Exemple, 220
Création
Nouveau pool de stockage mis en miroir (`zpool split`)
Exemple, 71
Pool de stockage avec périphérique de cache (exemple), 57
Pool de stockage RAID-Z à double parité (`zpool create`, commande)
Exemple, 54
Pool de stockage RAID-Z à parité simple (`zpool create`)
Exemple, 54
Pool de stockage RAID-Z à triple parité (`zpool create`, commande)
Exemple, 54
Pool de stockage ZFS
Description, 52
Pool de stockage ZFS (`zpool create`)
Exemple, 39, 52
Pool de stockage ZFS avec périphériques de journalisation (exemple), 55
Pool de stockage ZFS mis en miroir (`zpool create`)
Exemple, 52
Pools root de remplacement
Exemple, 291
Système de fichiers ZFS, 42
Description, 180
Exemple, 180
Système de fichiers ZFS de base (`zpool create`)
Exemple, 39
Création *(Suite)*
Volume ZFS
Exemple, 281
`creation` (propriété), description, 186

D

Définition
ACL sur un fichier ZFS (mode détaillé)
Description, 251
`atime`, propriété ZFS
exemple, 200
`compression`, propriété
Exemple, 42
`mountpoint`, propriété, 42
Point de montage ZFS (`zfs set mountpoint`)
Exemple, 207
Points de montage hérités
Exemple, 207
`quota`, propriété (exemple), 43
Quota d'un système de fichiers ZFS (`zfs set quota`)
Exemple, 213
Quota ZFS
(exemple), 201
Réservation de système de fichiers ZFS
Exemple, 216
`sharenfs`, propriété
Exemple, 42
Définition de propriétés ZFS
`aclinherit`, 184
`canmount`, 185
`checksum`, 185
`devices`, 186
`mountpoint`, 186
`recordsize`, 188
`reservation`, 189
`shareiscsi`, 189
`sharenfs`, 190
`snapdir`, 190
`version`, 191
`zoned`, 192
Délégation
Autorisations (exemple), 272

Délégation *(Suite)*
Jeu de données à une zone non globale
Exemple, 286
Délégation d'autorisations, `zfs allow`, 271
Délégation d'autorisations à un groupe, Exemple, 273
Délégation d'autorisations à un utilisateur individuel, Exemple, 272
Démontage
Système de fichiers ZFS
Exemple, 209
Dépannage
Détection de problèmes éventuels (`zpool status -x`), 298
Déterminer si un périphérique peut être remplacé
Description, 307
Echecs ZFS, 293
Périphérique manquant (UNAVAIL), 303
Périphériques endommagés, 315
Problème d'identification, 297
Remplacement d'un périphérique (`zpool replace`)
Exemple, 314
Remplacement d'un périphérique manquant
Exemple, 301
Réparation d'un système qui ne peut être initialisé
Description, 325
Réparation d'une configuration ZFS endommagée, 325
Réparation de dommages au niveau d'un pool
Description, 324
Destruction
Clone ZFS, exemple, 228
Instantané ZFS
Exemple, 221
Pool de stockage ZFS
Description, 52
Pool de stockage ZFS (`zpool destroy`)
Exemple, 63
Système de fichiers ZFS
Exemple, 181
Système de fichiers ZFS comportant des systèmes dépendants
Exemple, 182
Détection
Niveaux de réplication incohérents
Exemple, 61
Périphérique en cours d'utilisation
Exemple, 60
Détermination
Type de panne de périphérique
Description, 305
Déterminer
Remplacement d'un périphérique
Description, 307
`devices`, propriété, Description, 186
Différences entre le système de fichiers ZFS et les systèmes de fichiers standard, Nouveau modèle d'ACL standard, 35
Différences entre un système de fichiers classique et ZFS, Granularité d'un système de fichiers, 32
Différences existant entre les systèmes de fichiers traditionnels et ZFS
Comportement d'espace saturé, 34
Comptabilisation de l'espace ZFS, 33
Gestion d'un volume traditionnel, 35
Montage d'un système de fichiers ZFS, 34
Disque, Composant de pool de stockage ZFS, 46
Disque entier, Composant de pool de stockage ZFS, 46
Disque hot spare
Création
Exemple, 80
Description
Exemple, 80
Données
Altération, 318
Altération identifiée (`zpool status -v`)
Exemple, 300
Nettoyage
Exemple, 316
Réargenture
Description, 317
Réparation, 316
Validation (nettoyage), 316
`dumpadm`, commande, Activation d'un périphérique de vidage, 161

E

Echecs, 293
Effacement
  Périphérique d'un pool de stockage ZFS (`zpool clear`)
    Description, 77
Enregistrement
  Données d'un système de fichiers ZFS (`zfs send`)
    Exemple, 232
  Vidage sur incident
    `savecore`, commande, 161
Entrelacement dynamique
  Description, 51
  Fonction de pool de stockage, 51
Envoi et réception
  Données d'un système de fichiers ZFS
    Description, 229
Etiquette EFI
  Description, 46
  Interaction avec ZFS, 46
`exec`, propriété, Description, 186
Exigences matérielles et logicielles, 38
Exportation
  Pool de stockage ZFS
    Exemple, 101

F

`failmode` (propriété), description, 87
Fichiers, Composants de pools de stockage ZFS, 48
Fonctionnalités des réplications ZFS, Mise en miroir ou RAID-Z, 49
`free` (propriété), description, 88

G

Gestion d'un volume traditionnel, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 35
Granularité d'un système de fichiers, Différences entre un système de fichiers classique et ZFS, 32

H

`health`, propriété (description), 88
Héritage
  Propriétés ZFS (`zfs inherit`)
    Description, 201
Hiérarchie d'un système de fichiers, Création, 41

I

Identification
  Pool de stockage ZFS à importer (`zpool import -a`)
    Exemple, 102
  Stockage requis, 39
  Type d'altération de données (`zpool status -v`)
    Exemple, 320
Importation
  Pool de stockage ZFS
    Exemple, 104
  Pool de stockage ZFS, à partir de répertoires alternatifs (`zpool import -d`)
    Exemple, 103
  Pools root de remplacement
    Exemple, 291
Initialisation
  Environnement d'initialisation ZFS avec `boot -L` et `boot -Z` sur un système SPARC, 165
  Système de fichiers root, 162
Installation
  Système de fichiers root ZFS
    Configuration requise, 113
    Fonctions, 112
    Installation initiale, 116
    Installation JumpStart, 128
Installation des blocs d'initialisation
  `installboot` et `installgrup`
    Exemple, 163
Installation initiale du système de fichiers root ZFS, Exemple, 117
Installation JumpStart
  Système de fichiers root
    Exemples de profils, 130
    Problèmes, 131

Instantané
Accès
Exemple, 223
Comptabilisation d'espace, 224
Création
Exemple, 220
Définition, 30
Destruction
Exemple, 221
Fonction, 219
Renommer
Exemple, 222
Restauration
Exemple, 225

J

Jeu d'autorisations défini, 267
Jeu de données
Définition, 29
Description, 180
Jeux de données, types, Description, 199
Journal d'intention ZFS (ZIL), Description, 55

L

Lecture seule, propriétés ZFS
`compression`, 185
Description, 192
`origin`, 187
`referenced`, 188
`type`, 190
Liste
Descendants des systèmes de fichier ZFS (Exemple de), 198
Informations sur le pool ZFS, 40
Pool de stockage ZFS
Description, 88
Pools de stockage ZFS
Exemple, 89
Propriétés ZFS (`zfs list`)
Exemple, 202
Liste *(Suite)*
Propriétés ZFS par valeur de source
Exemple, 204
Propriétés ZFS pour l'exécution de scripts
Exemple, 205
Système de fichiers ZFS
Exemple, 198
Système de fichiers ZFS (`zfs list`)
Exemple, 43
Système de fichiers ZFS sans l'en-tête
Exemple, 200
Types de systèmes de fichiers ZFS
Exemple, 199
`listshares`, propriété (description), 88
`listsnapshots` (propriété), description, 88
`luactivate`
Système de fichiers root
Exemple, 136
`lucreate`
Environnement d'initialisation ZFS à partir d'un environnement d'initialisation ZFS
Exemple, 138
`lucreate`, commande
Migration de système de fichiers root
Exemple, 135

M

Migration
Système de fichiers root UFS vers système de fichiers root ZFS
Live Upgrade, 132
Système de fichiers root UFS vers un système de fichiers root ZFS
Problèmes, 133
Migration d'un pool de stockage ZFS, Description, 100
Miroir, Définition, 30
Mise à niveau
Pool de stockage ZFS
Description, 109
Systèmes de fichiers ZFS
Description, 218

Mise en ligne d'un périphérique
  Pool de stockage ZFS (zpool online)
    Exemple, 76
Mise en ligne et hors ligne de périphérique
  Pool de stockage ZFS
    Description, 74
Mise hors ligne d'un périphérique (zpool offline)
  Pool de stockage ZFS
    Exemple, 75
Mode de panne
  Données altérées, 318
  Périphérique manquant (UNAVAIL), 303
Mode de propriétés d'ACL, aclmode, 184
Modèle d'ACL Solaris, Différences entre le système de fichiers ZFS et les systèmes de fichiers standard, 35
Modes d'échec, Périphériques endommagés, 315
Modification
  ACL triviale dans un fichier ZFS (mode détaillé) (Exemple), 252
Modification du nom
  Système de fichiers ZFS
    Exemple, 182
Montage
  Système de fichiers ZFS
    Exemple, 208
Montage d'un système de fichiers ZFS, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 34
Mots-clés de profil JumpStart, Système de fichiers root ZFS, 128
mounted (propriété), Description, 186
mountpoint, propriété, Description, 186

N

Nettoyage
  Exemple, 316
  Validation des données, 316
Nettoyage et réargenture, Description, 317
Niveaux de réplication incohérents
  Détection
    Exemple, 61
Notification
  périphérique reconnecté dans ZFS(zpool online)
    Exemple, 304

O

Oracle Solaris Live Upgrade
  Migration d'un système de fichiers root, 132
  Migration de système de fichiers root
    Exemple, 135
  Problèmes de migration d'un système de fichiers root, 133
origin, propriété, Description, 187

P

Package de flux
  Récursif, 231
  Réplication, 231
Package de flux de réplication, 231
Package de flux récursif, 231
Partage
  Système de fichiers ZFS
    Description, 210
    Exemple, 211
Partage d'un pool de stockage mis en miroir (zpool split)
    Exemple, 71
Périphérique de cache
  Considérations d'utilisation, 57
  Création d'un pool (exemple), 57
Périphérique de journalisation mis en miroir, ajout, Exemple, 66
Périphérique de swap et de vidage, Problèmes, 158
Périphérique en cours d'utilisation
  Détection
    Exemple, 60
Périphérique virtuel
  Composant de pools de stockage ZFS, 59
  Définition, 31
Périphériques de cache , suppression, Exemple, 68
Périphériques de cache, ajout, Exemple, 68

Périphériques de journalisation distincts, considérations pour l'utilisation, 55
Périphériques de journalisation mis en miroir, Création d'un pool de stockage ZFS (exemple), 55
Périphériques de swap et de vidage
  Ajustement de la taille, 159
  Description, 158
Point de montage
  Héritage, 206
  Par défaut pour un système de fichiers ZFS, 180
  Valeur par défaut des pools de stockage ZFS, 62
Points de montage
  Automatique, 206
  Gestion dans ZFS
    Description, 206
Pool, Définition, 30
Pool de stockage mis en miroir (`zpool create`), Exemple, 52
Pool de stockage ZFS
  Affichage de l'état de maintenance
    Exemple, 96
  Affichage de l'état détaillé du fonctionnement
    Exemple, 97
  Affichage de l'état fonctionnel, 95
  Ajout de périphérique (`zpool add`)
    Exemple, 64
  Altération de données identifiée (`zpool status -v`)
    Exemple, 300
  Composants, 45
  Configuration en miroir, description, 49
  Configuration RAID-Z, création (`zpool create`)
    Exemple, 54
  Configuration RAID-Z, description, 49
  Connexion de périphériques (`zpool attach`)
    Exemple, 69
  Création (`zpool create`)
    Exemple, 52
  Création d'une configuration mise en miroir (`zpool create`)
    Exemple, 52
  Destruction (`zpool destroy`)
    Exemple, 63
  Détection de problèmes éventuels (`zpool status -x`)
Pool de stockage ZFS, Détection de problèmes éventuels (`zpool status -x`) *(Suite)*
    Description, 298
  Détermination du type de panne de périphérique
    Description, 305
  Déterminer si un périphérique peut être remplacé
    Description, 307
  Données altérées
    Description, 318
  Entrelacement dynamique, 51
  Exportation
    Exemple, 101
  Identification du type d'altération de données (`zpool status -v`)
    Exemple, 320
  Identification pour l'importation (`zpool import -a`)
    Exemple, 102
  Importation à partir de répertoires alternatifs (`zpool import -d`)
    Exemple, 103
  Informations globales d'état des pools pour la résolution de problèmes
    Description, 298
  Liste
    Exemple, 89
  Migration
    Description, 100
  Miroir
    Définition, 30
  Mise en ligne et hors ligne de périphérique
    Description, 74
  Mise hors ligne d'un périphérique (`zpool offline`)
    Exemple, 75
  Nettoyage de données
    Exemple, 316
  Nettoyage des données
    Description, 316
  Nettoyage et réargenture de données
    Description, 317
  Notification d'un périphérique reconnecté dans ZFS (`zpool online`)
    Exemple, 304
  Périphérique manquant (UNAVAIL)
    Description, 303

Pool de stockage ZFS *(Suite)*
Périphérique virtuel, 59
Point de montage par défaut, 62
Pool
Définition, 30
Problème d'identification
Description, 297
Profils de droits, 37
RAID-Z
Définition, 30
Réargenture
Définition, 30
Récupération d'un pool détruit
Exemple, 108
Remplacement d'un périphérique (`zpool replace`)
Exemple, 77
Remplacement de périphérique (`zpool replace`)
Exemple, 308
Réparation d'un fichier ou d'un répertoire endommagé
Description, 321
Réparation d'un système qui ne peut être initialisé
Description, 325
Réparation d'une configuration ZFS endommagée, 325
Réparation de dommages au niveau d'un pool
Description, 324
Réparation des données
Description, 316
Script de sortie de pool de stockage
Exemple, 90
Séparation des périphériques (`zpool detach`)
Exemple, 71
Statistiques d'E/S à l'échelle du pool
Exemple, 93
Statistiques d'E/S vdev
Exemple, 94
Suppression d'un périphérique
Exemple, 77
Suppression des erreurs de périphérique (`zpool clear`)
Exemple, 306
Test (`zpool create -n`)
Exemple, 62
Pool de stockage ZFS *(Suite)*
Utilisation de disques entiers, 46
Validation des données
Description, 316
Version
Description, 337
Pool de stockage ZFS (`zpool online`)
Mise en ligne d'un périphérique
Exemple, 76
Pool ZFS, propriété, `version`, 88
Pools de stockage ZFS
Affichage du processus de réargenture
Exemple, 314
Echecs, 293
Importation
Exemple, 104
Messages d'erreur système
Description, 296
Mise à niveau
Description, 109
Partage d'un pool de stockage mis en miroir (`zpool split`)
Exemple, 71
Périphériques endommagés
Description, 315
Pools root de remplacement, 290
Remplacement d'un périphérique manquant
Exemple, 301
Utilisation de fichiers, 48
Pools root de remplacement
Création
Exemple, 291
Description, 290
Importation
Exemple, 291
`primarycache`, propriété, Description, 187
Profils de droits, Gestion de systèmes de fichiers et pools de stockage ZFS, 37
Propriété `capacity`, description, 87
Propriété de pool ZFS
`allocated` (propriété), 86
`altroot`, 86
`autoreplace` (propriété), 86
`capacity`, 87

Propriété de pool ZFS *(Suite)*
delegation, 87
failmode (propriété), 87
guid, 88
health, 88
listsnapshots (propriété), 88
size, 88
propriété delegation, désactivation, 268
Propriété delegation, description, 87
Propriété guid, description, 88
Propriété size, description, 88
Propriété ZFS
exec, 186
shareiscsi, 189
Propriété ZFS pouvant être définie, exec, 186
Propriétés de pool ZFS
bootfs, 86
cachefile (propriété), 87
free (propriété), 88
listsharess, 88
Propriétés en lecture seule de ZFS,
usedbychildren, 190
Propriétés ZFS
aclinherit, 184
aclmode, 184
atime, 184
available, 184
canmount, 185
Description détaillée, 195
checksum, 185
compression, 185
compressratio, 185
copies, 186
creation, 186
Définissables, 193
Description, 183
Description des propriétés héritées, 183
devices, 186
Gestion au sein d'une zone
Description, 288
Héritées, description, 183
mounted, 186
mountpoint, 186
origin, 187
Propriétés ZFS *(Suite)*
Propriétés définies par l'utilisateur
Description détaillée, 196
quota, 187
read-only, 187
recordsize, 188
Description détaillée, 195
referenced, 188
refquota, 188
refreservation, 188
reservation, 189
secondarycache, 187, 189
setuid, 189
sharenfs, 190
snapdir, 190
type, 190
used, 190
Description détaillée, 193
usedbychildren, 190
usedbydataset, 190
usedbyrefreservation, 191
usedbysnapshots, 191
version, 191
volblocksize, 191
volsize, 191
Description détaillée, 196
xattr, 192
zoned, 192
zoned, propriété
Description détaillée, 289
Propriétés ZFS définies par l'utilisateur
Description détaillée, 196
Exemple, 196
Propriétés ZFS définissables
aclmode, 184
atime, 184
canmount
Description détaillée, 195
compression, 185
copies, 186
Description, 193
primarycache, 187
quota, 187
read-only, 187

Propriétés ZFS définissables *(Suite)*
recordsize
Description détaillée, 195
refquota, 188
refreservation, 188
secondarycache, 189
setuid, 189
used
Description détaillée, 193
volblocksize, 191
volsize, 191
Description détaillée, 196
xattr, 192
Propriétés ZFS en lecture seule
available, 184
creation, 186
mounted, 186
used, 190
usedbydataset, 190
usedbyrefreservation, 191
usedbysnapshots, 191

Q

quota, propriété, Description, 187
Quotas et réservations, Description, 212

R

RAID-Z, Définition, 30
read-only (propriété), Description, 187
Réargenture, Définition, 30
Réception
Données de système de fichiers ZFS (zfs receive)
Exemple, 233
recordsize, propriété, Description, 188
recordsize (propriété), Description détaillée, 195
Récupération
Pool de stockage ZFS détruit
Exemple, 108
referenced, propriété, Description, 188
refquota, propriété, Description, 188
refreservation (propriété), Description, 188
Remplacement
Périphérique (zpool replace)
Exemple, 77, 308, 314
Périphérique manquant
Exemple, 301
Renommer
Instantané ZFS
Exemple, 222
Réparation
Configuration ZFS endommagée
Description, 325
Dommages au niveau d'un pool
Description, 324
Réparation d'un fichier ou d'un répertoire endommagé
Description, 321
Système qui ne peut être initialisé
Description, 325
reservation, propriété, Description, 189
Résolution de problèmes
Altération de données identifiée (zpool status -v)
Exemple, 300
Détermination du type d'altération de données (zpool status -v)
Exemple, 320
Détermination du type de panne de périphérique
Description, 305
Informations globales d'état des pools
Description, 298
Notification d'un périphérique reconnecté dans ZFS (zpool online)
Exemple, 304
Rapport syslog de messages d'erreur ZFS, 296
Remplacement de périphérique (zpool replace)
Exemple, 308
Réparation d'un fichier ou d'un répertoire endommagé
Description, 321
Suppression des erreurs de périphérique (zpool clear)
Exemple, 306
Restauration
ACL triviale sur un fichier ZFS (mode détaillé)
Exemple, 254

Restauration *(Suite)*
Instantané ZFS
Exemple, 225

S

`savecore`, commande, Enregistrement de vidage sur incident, 161
script
Sortie de pool de stockage ZFS
Exemple, 90
`secondarycache`, propriété, Description, 189
Sémantique transactionnelle, Description, 27
Séparation
Périphérique, d'un pool de stockage ZFS (`zpool detach`)
Exemple, 71
`setuid` (propriété), description, 189
`shareiscsi` (propriété), Description, 189
`sharenfs`, propriété
Description, 190, 210
`snapdir`, propriété, Description, 190
Somme de contrôle, Définition, 29
Somme de contrôle de données, Description, 27
Stockage requis, Identification, 39
Stockage sur pool, Description, 26
Suppression
Erreurs de périphérique (`zpool clear`)
Exemple, 306
Périphériques de cache (exemple), 68
Suppression d'autorisations, `zfs unallow`, 272
Suppression d'un périphérique
Pool de stockage ZFS
Exemple, 77
Système de fichiers, Définition, 29
Système de fichiers ZFS
ACL dans un fichier ZFS
Description détaillée, 249
ACL dans un répertoire ZFS
Description détaillée, 250
Administration simplifiée
Description, 28
Ajout d'un système de fichiers ZFS à une zone non globale
Système de fichiers ZFS, Ajout d'un système de fichiers ZFS à une zone non globale *(Suite)*
Exemple, 286
Ajout d'un volume ZFS à une zone non globale (exemple), 287
Annulation du partage
exemple, 211
Clone, 228
Définition, 29
Description, 227
Remplacement d'un système de fichiers (exemple), 228
Comptabilisation d'espace d'instantané, 224
Configuration d'ACL dans des fichiers ZFS
Description, 248
Configuration d'ACL dans un fichier ZFS (mode compact)
Description, 261
Exemple, 262
Configuration requise pour l'installation et Oracle Solaris Live Upgrade, 113
Convention d'attribution de nom de composant, 31
Création
Exemple, 180
Création d'un volume ZFS
Exemple, 281
Création de clone, 227
Définition d'un point de montage (`zfs set mountpoint`)
Exemple, 207
Définition d'un point de montage hérité
Exemple, 207
Définition d'une réservation
Exemple, 216
Définition de la propriété `atime`
Exemple, 200
Définition de la propriété `quota`
(exemple), 201
Définition des ACL sur un fichier ZFS (mode détaillé)
Description, 251
Délégation d'un jeu de données à une zone non globale
Exemple, 286

Système de fichiers ZFS *(Suite)*
Démontage
Exemple, 209
Description, 179
Destruction
Exemple, 181
Destruction avec les systèmes dépendants
Exemple, 182
Enregistrement de flux de données (`zfs send`)
Exemple, 232
Envoi et réception
Description, 229
Gestion de propriété au sein d'une zone
Description, 288
Gestion des points de montage
Description, 206
Gestion des points de montage hérités
Description, 206
Héritage d'ACL dans un fichier ZFS (mode détaillé)
Exemple, 255
système de fichiers ZFS
Initialisation d'un environnement d'initialisation ZFS avec `boot -L` et `boot -Z`
Exemple SPARC, 165
Système de fichiers ZFS
Initialisation d'un système de fichiers root
Description, 162
Installation d'un système de fichiers root, 112
Installation initiale d'un système de fichiers root ZFS, 116
Installation JumpStart du système de fichiers root, 128
Instantané
Définition, 30
Description, 219
Renommer, 222
Jeu de données
Définition, 29
Liste
Exemple, 198
Liste de propriétés (`zfs list`)
Exemple, 202
Liste des propriétés pour l'exécution de scripts
Exemple, 205
Système de fichiers ZFS *(Suite)*
Liste des types
Exemple, 199
Liste sans en-tête
Exemple, 200
Migration d'un système de fichiers root avec Oracle Solaris Live Upgrade, 132
Migration de système de fichiers root avec Oracle Solaris Live Upgrade
Exemple, 135
Modification du nom
Exemple, 182
Montage
Exemple, 208
Partage
Description, 210
Exemple, 211
Périphérique de swap et de vidage
Problème, 158
Périphériques de swap et de vidage
Ajustement de la taille, 159
Description, 158
Point de montage par défaut
Exemple, 180
Profils de droits, 37
Restauration d'une ACL triviale sur un fichier ZFS (mode détaillé)
Exemple, 254
Sémantique transactionnelle
Description, 27
Somme de contrôle
Définition, 29
Somme de contrôle de données
Description, 27
Stockage sur pool
Description, 26
Système de fichiers
Définition, 29
Types de jeux de données
Description, 199
Utilisation d'un système Solaris avec zones installées
Description, 285

système de fichiers ZFS
Version
Description, 337
Système de fichiers ZFS
Volume
Définition, 31
Système de stockage ZFS
Périphérique virtuel
Définition, 31
Systèmes de fichiers root ZFS, Problèmes de migration d'un système de fichiers root, 133
Systèmes de fichiers ZFS
Description, 26
Gestion des points de montage automatiques, 206
Héritage d'une propriété de (`zfs inherit`)
Exemple, 201
Instantané
Accès, 223
Création, 220
Destruction, 221
Restauration, 225
Liste des descendants
(Exemple de), 198
Liste des propriétés par valeur de source
Exemple, 204
Mise à niveau
Description, 218
Modification d'ACL triviale dans un fichier ZFS
(mode détaillé)
(Exemple), 252
Réception de flux de données (`zfs receive`)
Exemple, 233
Systèmes de fichiers ZFS (`zfs set quota`)
Définition d'un quota
Exemple, 213

T

Terminologie
Clone, 29
Instantané, 30
Jeu de données, 29
Miroir, 30
Périphérique virtuel, 31
Terminologie *(Suite)*
Pool, 30
RAID-Z, 30
Réargenture, 30
Somme de contrôle, 29
Système de fichiers, 29
Volume, 31
Test
Création de pool de stockage ZFS (`zpool create -n`)
Exemple, 62
`type`, propriété, Description, 190

U

`used` (propriété)
Description, 190
Description détaillée, 193
`usedbychildren`, propriété, Description, 190
`usedbydataset` Propriété, Description, 190
`usedbyrefreservation` Propriété, Description, 191
`usedbysnapshots`Propriété, Description, 191

V

`version` (propriété), Description, 191
`version` propriété, description, 88
Version ZFS
Fonction ZFS et SE Solaris
Description, 337
`Vidage sur incident`, Enregistrement, 161
`volblocksize` (propriété), description, 191
`volsize`, propriété, Description, 191
`volsize` (propriété), Description détaillée, 196
Volume, Définition, 31
Volume ZFS, Description, 281

X

`xattr`, Description, 192

Z

ZFS, Propriétés, Lecture seule, 192
zfs allow
  Affichage des autorisations déléguées, 276
  Description, 271
zfs create
  Description, 180
  Exemple, 42, 180
zfs destroy, Exemple, 181
zfs destroy -r, Exemple, 182
zfs get, Exemple, 202
zfs get -H -o, commande, Exemple, 205
zfs get -s, Exemple, 204
zfs inherit, Exemple, 201
zfs list
  Exemple, 43, 198
zfs list -H, Exemple, 200
zfs list -r, (Exemple de), 198
zfs list -t, Exemple, 199
zfs mount, Exemple, 208
zfs promote, Promotion d'un clone (exemple), 228
zfs receive, Exemple, 233
zfs rename, Exemple, 182
zfs send, Exemple, 232
zfs set atime, commande, Exemple, 200
zfs set compression, Exemple, 42
zfs set mountpoint
  Exemple, 42, 207
zfs set mountpoint=legacy, Exemple, 207
zfs set quota, (exemple), 201
zfs set quota
  Exemple, 43, 213
zfs set reservation, Exemple, 216
zfs set sharenfs, Exemple, 42
zfs set sharenfs=on, Exemple, 211
zfs unallow, Description, 272
zfs unmount, Exemple, 209
zfs upgrade, 218
Zone
  Ajout d'un système de fichiers ZFS à une zone non globale
    Exemple, 286
  Délégation d'un jeu de données à une zone non globale
Zone, Délégation d'un jeu de données à une zone non globale *(Suite)*
    Exemple, 286
  Gestion de propriétés ZFS au sein d'une zone
    Description, 288
  Utilisation de systèmes de fichiers ZFS
    Description, 285
zoned, propriété
  Description, 192
  Description détaillée, 289
Zones
  Ajout d'un volume ZFS à une zone non globale (exemple), 287
  zoned, propriété
    Description détaillée, 289
zpool add, Exemple, 64
zpool attach, Exemple, 69
zpool clear
  Description, 77
  Exemple, 77
zpool create
  Exemple, 39, 40
  Pool de base
    Exemple, 52
  Pool de stockage mis en miroir
    Exemple, 52
  Pool de stockage RAID-Z
    Exemple, 54
zpool create -n, Test (exemple), 62
zpool destroy, Exemple, 63
zpool detach, Exemple, 71
zpool export, Exemple, 101
zpool import -a, Exemple, 102
zpool import -D, Exemple, 108
zpool import -d, Exemple, 103
zpool import *name*, Exemple, 104
zpool iostat, pool complet, exemple, 93
zpool iostat -v, vdev, exemple, 94
zpool list
  Description, 88
  Exemple, 40, 89
zpool list -Ho name, Exemple, 90
zpool offline, Exemple, 75
zpool online, Exemple, 76

`zpool replace`, Exemple, 77
`zpool split`, Exemple, 71
`zpool status -v`, Exemple, 97
`zpool status -x`, commande, Exemple, 96
`zpool upgrade`, 109