



Guide d'administration Solaris ZFS



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Référence : 820-2315-12
Octobre 2008

Sun Microsystems, Inc. détient les droits de propriété intellectuelle de la technologie utilisée par le produit décrit dans le présent document. En particulier, et sans limitation, ces droits de propriété intellectuelle peuvent inclure des brevets américains ou dépôts de brevets en cours d'homologation aux États-Unis et dans d'autres pays.

Droits du gouvernement américain – logiciel commercial. Les utilisateurs gouvernementaux sont soumis au contrat de licence standard de Sun Microsystems, Inc. et aux dispositions du Federal Acquisition Regulation (FAR, règlements des marchés publics fédéraux) et de leurs suppléments.

Cette distribution peut contenir des éléments développés par des tiers.

Des parties du produit peuvent être dérivées de systèmes Berkeley-BSD, sous licence de l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays, sous licence exclusive de X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java (tasse de café), docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont constitués selon une architecture développée par Sun Microsystems, Inc. Legato NetWorker est une marque commerciale ou une marque déposée de Legato Systems, Inc.

L'interface utilisateur graphique OPEN LOOK et SunTM a été développée par Sun Microsystems, Inc. pour ses utilisateurs et détenteurs de licence. Sun reconnaît le travail précurseur de Xerox en matière de recherche et de développement du concept d'interfaces utilisateur visuelles ou graphiques pour le secteur de l'informatique. Sun détient une licence Xerox non exclusive sur l'interface utilisateur graphique Xerox. Cette licence englobe également les détenteurs de licences Sun qui implémentent l'interface utilisateur graphique OPEN LOOK et qui, en outre, se conforment aux accords de licence écrits de Sun.

Les produits cités dans la présente publication et les informations qu'elle contient sont soumis à la législation américaine relative au contrôle sur les exportations et, le cas échéant, aux lois sur les importations ou exportations dans d'autres pays. Il est strictement interdit d'employer ce produit conjointement à des missiles ou armes biologiques, chimiques, nucléaires ou de marine nucléaire, directement ou indirectement. Il est strictement interdit d'effectuer des exportations et réexportations vers des pays soumis à l'embargo américain ou vers des entités identifiées sur les listes noires des exportations américaines, notamment les individus non autorisés et les listes nationales désignées.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, REPRÉSENTATIONS ET GARANTIES EXPRESSES OU TACITES, Y COMPRIS TOUTE GARANTIE IMPLICITE RELATIVE À LA COMMERCIALISATION, L'ADÉQUATION À UN USAGE PARTICULIER OU LA NON-VIOLATION DE DROIT, SONT FORMELLEMENT EXCLUES. CETTE EXCLUSION DE GARANTIE NE S'APPLIQUERAIT PAS DANS LA MESURE OÙ ELLE SERAIT TENUE JURIDIQUEMENT NULLE ET NON AVENUE.

Table des matières

Préface	11
1 Système de fichiers ZFS (présentation)	15
Nouveautés de ZFS	15
Prise en charge de l'installation et de l'initialisation de ZFS	16
Rétablissement d'un jeu de données sans démontage	16
Améliorations apportées à la commande <code>zfs send</code>	17
Quotas et réservations ZFS pour les données de système de fichiers uniquement	18
Propriétés de pool de stockage ZFS	18
Améliorations apportées à l'historique des commandes ZFS (<code>zpool history</code>)	19
Mise à niveau des systèmes de fichiers ZFS (<code>zfs upgrade</code>)	20
Administration déléguée de ZFS	21
Configuration de périphériques de journalisation ZFS distincts	21
Création de jeux de données ZFS intermédiaires	23
Améliorations apportées à la connexion à chaud à ZFS	23
Renommage récursif d'instantanés ZFS (<code>zfs rename -r</code>)	24
La compression GZIP est disponible pour ZFS	25
Stockage de plusieurs copies de données utilisateur ZFS	25
Amélioration de la sortie de la commande <code>zpool status</code>	26
Améliorations de ZFS et Solaris iSCSI	27
Historique de commande ZFS (<code>zpool history</code>)	27
Améliorations de propriétés ZFS	28
Affichage de la totalité des informations de systèmes de fichiers ZFS	29
Nouvelle option <code>zfs receive -F</code>	29
Instantanés ZFS récursifs	30
Double Parité RAID-Z (<code>raidz2</code>)	30
Disques hot spare pour périphériques de pool de stockage ZFS	30
Remplacement d'un système de fichiers ZFS par un clone ZFS (<code>zfs promote</code>)	31

Mise à niveau des pools de stockage ZFS (zpool upgrade)	31
Commandes de sauvegarde et de rétablissement ZFS renommées	31
Récupération de pools de stockage détruits	31
Intégration de ZFS au gestionnaire de pannes	32
Nouvelle commande <code>zpool clear</code>	32
Format NFSv4 ACL compact	32
Outil de contrôle de système de fichiers (<code>fsstat</code>)	33
Gestion Web ZFS	33
Description de ZFS	34
Stockage ZFS mis en pool	34
Sémantique transactionnelle	35
Sommes de contrôle et données d'autorétablissement	35
Évolutivité inégale	36
Instantanés ZFS	36
Administration simplifiée	36
Terminologie ZFS	37
Exigences d'attribution de noms de composants ZFS	40
2 Guide de démarrage de ZFS	41
Exigences et recommandations en matière de matériel et de logiciel ZFS	41
Création d'un système de fichiers ZFS basique	42
Création d'un pool de stockage ZFS	43
▼ Identification des exigences de stockage du pool de stockage ZFS	43
▼ Création d'un pool de stockage ZFS	44
Création d'une hiérarchie de systèmes de fichiers ZFS	44
▼ Détermination de la hiérarchie du système de fichiers ZFS	45
▼ Création de systèmes de fichiers ZFS	46
3 Différences entre ZFS et les systèmes de fichiers classiques	49
Granularité du système de fichiers ZFS	49
Comptabilisation de l'espace ZFS	50
Comportement d'espace saturé	50
Montage de système de fichiers ZFS	51
Gestion de volumes classique	51
Nouveau modèle ACL Solaris	51

4	Installation et initialisation d'un système de fichiers racine ZFS	53
	Installation et initialisation d'un système de fichiers racine ZFS (présentation)	53
	Fonctions d'installation de ZFS	54
	Configuration requise pour l'installation de Solaris et de Solaris Live Upgrade pour la prise en charge de ZFS	55
	Installation d'un système de fichiers racine ZFS (installation initiale)	57
	Installation d'un système de fichiers racine ZFS (installation JumpStart)	63
	Exemples de profils JumpStart de ZFS	63
	Mots clés JumpStart de ZFS	64
	Problèmes liés à l'installation JumpStart d'un ZFS	66
	Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Solaris Live Upgrade)	67
	Problèmes de migration d'un ZFS avec Solaris Live Upgrade	69
	Utilisation de Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS (sans zones)	70
	Utilisation de Solaris Live Upgrade pour migrer un système avec zones	75
	Prise en charge ZFS des périphériques de swap et de vidage	81
	Ajustement de la taille de vos périphériques de swap et de vidage ZFS	82
	Initialisation à partir d'un système de fichiers racine ZFS	83
	Initialisation à partir d'un disque alternatif d'un pool racine ZFS mis en miroir	83
	Initialisation à partir d'un système de fichiers racine ZFS sur un système SPARC	85
	Initialisation à partir d'un système de fichiers racine ZFS sur un système x86	87
	Résolution des problèmes de point de montage ZFS responsables de l'échec de l'initialisation	88
5	Gestion des pools de stockage ZFS	91
	Composants d'un pool de stockage ZFS	91
	Utilisation de disques dans un pool de stockage ZFS	91
	Utilisation de tranches dans un pool de stockage ZFS	93
	Utilisation de fichiers dans un pool de stockage ZFS	93
	Fonctions de réplication d'un pool de stockage ZFS	94
	Configuration de pool de stockage mis en miroir	94
	Configuration de pool de stockage RAID-Z	94
	Données d'autorétablissement dans une configuration redondante	96
	Entrelacement dynamique dans un pool de stockage	96
	Création et destruction de pools de stockage ZFS	96

Création d'un pool de stockage ZFS	97
Affichage des informations d'un périphérique virtuel de pool de stockage	100
Gestion d'erreurs de création de pools de stockage ZFS	101
Destruction de pools de stockage ZFS	105
Gestion de périphériques dans un pool de stockage ZFS	106
Ajout de périphériques à un pool de stockage	106
Connexion et séparation de périphériques dans un pool de stockage	109
Mise en ligne et mise hors ligne de périphériques dans un pool de stockage	111
Suppression des périphériques de pool de stockage	114
Remplacement de périphériques dans un pool de stockage	114
Désignation des disques hot spare dans le pool de stockage	116
Gestion des propriétés de pool de stockage ZFS	120
Requête d'état de pool de stockage ZFS	123
Affichage des informations de pools de stockage ZFS de base	123
Visualisation de statistiques d'E/S de pools de stockage ZFS	125
Détermination de l'état de maintenance des pools de stockage ZFS	127
Migration de pools de stockage ZFS	130
Préparatifs de migration de pool de stockage ZFS	130
Exportation d'un pool de stockage ZFS	131
Définition des pools de stockage disponibles pour importation	131
Recherche de pools de stockage ZFS dans d'autres répertoires	134
Importation de pools de stockage ZFS	134
Récupération de pools de stockage ZFS détruits	135
Mise à niveau de pools de stockage ZFS	137
6 Gestion des systèmes de fichiers ZFS	139
Création et destruction de systèmes de fichiers ZFS	140
Création d'un système de fichiers ZFS	140
Destruction d'un système de fichiers ZFS	141
Modification du nom d'un système de fichiers ZFS	142
Présentation des propriétés ZFS	143
Propriétés ZFS natives en lecture seule	150
Propriétés ZFS natives définies	151
Propriétés ZFS définies par l'utilisateur	154
Envoi de requêtes sur les informations des systèmes de fichiers ZFS	156

Affichage des informations de base des systèmes ZFS	156
Création de requêtes ZFS complexes	157
Gestion des propriétés ZFS	158
Définition des propriétés ZFS	158
Héritage des propriétés ZFS	159
Envoi de requêtes sur les propriétés ZFS	160
Montage et partage des systèmes de fichiers ZFS	164
Gestion des points de montage ZFS	164
Montage de système de fichiers ZFS	166
Utilisation de propriétés de montage temporaires	167
Démontage des systèmes de fichiers ZFS	168
Activation et annulation du partage des systèmes de fichiers ZFS	169
Quotas et réservations ZFS	170
Définitions de quotas sur les systèmes de fichiers ZFS	171
Définition de réservations sur les systèmes de fichiers ZFS	173
7 Utilisation des instantanés et des clones ZFS	177
Présentation des instantanés ZFS	177
Création et destruction d'instantanés ZFS	178
Affichage et accès des instantanés ZFS	180
Restauration d'un instantané ZFS	181
Présentation des clones ZFS	182
Création d'un clone ZFS	182
Destruction d'un clone ZFS	183
Remplacement d'un système de fichiers ZFS par un clone ZFS	183
Envoi et réception de données ZFS	184
Envoi d'un instantané ZFS	185
Réception d'un instantané ZFS	186
Envoi et réception de flux d'instantanés ZFS complexes	187
Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde	190
8 Utilisation des ACL pour la protection de fichiers ZFS	191
Nouveau modèle ACL Solaris	191
Descriptions de syntaxe pour la configuration des ACL	193
Héritage d'ACL	196

Modes de propriétés d'ACL	197
Configuration d'ACL dans des fichiers ZFS	198
Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé	200
Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé	206
Configuration et affichage d'ACL dans des fichiers ZFS en format compact	216
9 Administration déléguée de ZFS	221
Présentation de l'administration déléguée de ZFS	221
Désactivation des droits délégués de ZFS	222
Délégation de droits ZFS	222
Description de la syntaxe de la délégation de droits (<code>zfs allow</code>)	224
Suppression des droits délégués de ZFS (<code>zfs unallow</code>)	225
Utilisation de l'administration déléguée de ZFS	226
Affichage des droits ZFS délégués (exemples)	226
Délégation de droits ZFS (exemples)	227
Suppression de droits ZFS (exemples)	232
10 Sections avancées de ZFS	235
Volumes ZFS	235
Utilisation d'un volume ZFS en tant que périphérique de swap ou de dump	236
Utilisation d'un volume ZFS en tant que cible iSCSI Solaris	237
Utilisation de ZFS dans un système Solaris avec zones installées	238
Ajout de systèmes de fichiers ZFS à une zone non globale	239
Délégation de jeux de données à une zone non globale	240
Ajout de volumes ZFS à une zone non globale	240
Utilisation de pools de stockage ZFS au sein d'une zone	241
Gestion de propriétés ZFS au sein d'une zone	241
Explication de la propriété <code>zoned</code>	242
Utilisation de pools racine ZFS de remplacement	244
Création de pools racine de remplacement ZFS	244
Importation de pools racine de remplacement	244
Profils de droits ZFS	245

11	Résolution de problèmes et récupération de données ZFS	247
	Modes de panne ZFS	247
	Périphériques manquants dans un pool de stockage ZFS	248
	Périphériques endommagés dans un pool de stockage ZFS	248
	Données ZFS corrompue	248
	Vérification de l'intégrité des données ZFS	249
	Réparation de données	249
	Validation de données	249
	Contrôle du nettoyage de données ZFS	250
	Identification de problèmes dans ZFS	251
	Recherche de problèmes éventuels dans un pool de stockage ZFS	253
	Consultation de la sortie de <code>zpool status</code>	253
	Rapport système de messages d'erreur ZFS	256
	Réparation d'une configuration ZFS endommagée	257
	Réparation d'un périphérique manquant	257
	Reconnexion physique du périphérique	258
	Notification relative à la disponibilité de périphériques dans ZFS	258
	Réparation d'un périphérique endommagé	259
	Détermination du type de panne de périphérique	259
	Suppression des erreurs transitoires	260
	Remplacement d'un périphérique dans un pool de stockage ZFS	261
	Réparation de données endommagées	267
	Identification du type de corruption de données	268
	Réparation d'un fichier ou répertoire corrompu	269
	Réparation de dommages présents dans l'ensemble du pool de stockage ZFS	270
	Réparation d'un système impossible à réinitialiser	271
	Index	273

Préface

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

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

Remarque – Cette version de Solaris prend en charge les systèmes utilisant les architectures de processeur SPARC et x86 : UltraSPARC®, SPARC64, AMD64, Pentium et Xeon EM64T. Les systèmes pris en charge sont répertoriés dans *la liste de compatibilité matérielle de Solaris* disponible à l'adresse <http://www.sun.com/bigadmin/hcl>. Ce document présente les différences d'implémentation en fonction des divers types de plates-formes.

Dans ce document, les termes x86 ci-dessous ont les significations suivantes :

- “x86” désigne la famille des produits compatibles x86 64 bits et 32 bits.
- “x64” désigne des informations 64 bits spécifiques relatives aux systèmes AMD64 ou EM64T.
- “x86 32 bits” désigne des informations 32 bits spécifiques relatives aux systèmes x86.

Pour connaître les systèmes pris en charge, reportez-vous à la *liste de compatibilité matérielle de Solaris 10*.

Utilisateurs de ce manuel

Ce guide est destiné à toute personne souhaitant configurer et gérer des systèmes de fichiers ZFS Solaris. Il est préférable d'être préalablement expérimenté en matière d'utilisation du système d'exploitation Solaris ou autre version UNIX®.

Organisation de ce document

Le tableau suivant décrit les chapitres de ce document.

Chapitre	Description
Chapitre1, “Système de fichiers ZFS (présentation)”	Présente ZFS, ses fonctionnalités et ses avantages. Il aborde également des concepts de base, ainsi que la terminologie.
Chapitre2, “Guide de démarrage de ZFS”	Décrit les instruction pas à pas d'une configuration ZFS simple, avec pools et systèmes de fichiers simples. Ce chapitre indique également le matériel et logiciels requis pour la création de systèmes de fichiers ZFS.
Chapitre3, “Différences entre ZFS et les systèmes de fichiers classiques”	Identifie les fonctionnalités importantes qui différencie ZFS des systèmes de fichiers classiques. La compréhension de ces différences clés permet d'éviter les confusions lors de l'utilisation d'outils classiques en interaction avec ZFS.
Chapitre4, “Installation et initialisation d'un système de fichiers racine ZFS”	Décrit la procédure d'installation et d'initialisation d'un système de fichiers ZFS. La migration d'un système de fichiers racine UFS vers un système de fichiers ZFS à l'aide de Solaris Live Upgrade est également abordée.
Chapitre5, “Gestion des pools de stockage ZFS”	Décrit en détail les méthodes de création et d'administration de pools de stockage.
Chapitre6, “Gestion des systèmes de fichiers 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.
Chapitre7, “Utilisation des instantanés et des clones ZFS”	Décrit les méthodes de création et d'administration d'instantanés ZFS et de clones.
Chapitre8, “Utilisation des ACL pour la protection de fichiers ZFS”	Décrit l'utilisation des listes de contrôle d'accès (ACL, Access Control List) pour la protection des fichiers ZFS en fournissant davantage de droits granulaires que les droits standard UNIX.
Chapitre9, “Administration déléguée de ZFS”	Décrit la méthode d'utilisation de l'administration déléguée de ZFS pour permettre aux utilisateurs ne disposant pas de privilèges appropriés d'effectuer des tâches d'administration ZFS.
Chapitre10, “Sections avancées de ZFS”	Décrit l'utilisation de volumes ZFS, l'utilisation de ZFS dans un système Solaris comprenant des zones, ainsi que les pools racine de remplacement.
Chapitre11, “Résolution de problèmes et récupération de données ZFS”	Décrit l'identification des modes de défaillance de ZFS et les solutions existantes. Les étapes de prévention de ces défaillances sont également abordées.

Documentation connexe

Vous trouverez des informations générales à propos de l'administration de systèmes Solaris dans les documents suivants :

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

Documentation, support et formation

Le site Web Sun fournit des informations sur les ressources supplémentaires suivantes :

- [documentation](http://www.sun.com/documentation/) ; (<http://www.sun.com/documentation/>)
- [support](http://www.sun.com/support/) ; (<http://www.sun.com/support/>)
- [formation](http://www.sun.com/training/). (<http://www.sun.com/training/>)

Conventions typographiques

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

TABLEAU P-1 Conventions typographiques

Type de caractères	Signification	Exemple
AaBbCc123	Noms des commandes, fichiers et répertoires, ainsi que messages système.	Modifiez votre fichier <code>.login</code> . Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers. <code>nom_machine%</code> Vous avez reçu du courrier.
AaBbCc123	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	<code>nom_machine%</code> su Mot de passe :
<i>aabbcc123</i>	Paramètre fictif : à remplacer par un nom ou une valeur réel(le).	La commande permettant de supprimer un fichier est <code>rm nom_fichier</code> .

TABLEAU P-1 Conventions typographiques (Suite)

Type de caractères	Signification	Exemple
<i>AaBbCc123</i>	Titres de manuel, nouveaux termes et termes importants.	Reportez-vous au chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie des éléments stockés localement. <i>N'enregistrez pas</i> 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 les invites système et les invites de superutilisateur UNIX par défaut des C shell, Bourne shell et Korn shell.

TABLEAU P-2 Invites de shell

Shell	Invite
C shell	nom_machine%
C shell pour superutilisateur	nom_machine#
Bourne shell et Korn shell	\$
Bourne shell et Korn shell pour superutilisateur	#

Systeme de fichiers ZFS (présentation)

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

Il contient les sections suivantes :

- “Nouveautés de ZFS” à la page 15
- “Description de ZFS” à la page 34
- “Terminologie ZFS” à la page 37
- “Exigences d'attribution de noms de composants ZFS” à la page 40

Nouveautés de ZFS

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

- “Prise en charge de l'installation et de l'initialisation de ZFS” à la page 16
- “Rétablissement d'un jeu de données sans démontage” à la page 16
- “Améliorations apportées à la commande `zfs send`” à la page 17
- “Quotas et réservations ZFS pour les données de système de fichiers uniquement” à la page 18
- “Propriétés de pool de stockage ZFS” à la page 18
- “Améliorations apportées à l'historique des commandes ZFS (`zpool history`)” à la page 19
- “Mise à niveau des systèmes de fichiers ZFS (`zfs upgrade`)” à la page 20
- “Administration déléguée de ZFS” à la page 21
- “Configuration de périphériques de journalisation ZFS distincts” à la page 21
- “Création de jeux de données ZFS intermédiaires” à la page 23
- “Améliorations apportées à la connexion à chaud à ZFS” à la page 23
- “Renommage récursif d'instantanés ZFS (`zfs rename -r`)” à la page 24
- “La compression GZIP est disponible pour ZFS” à la page 25
- “Stockage de plusieurs copies de données utilisateur ZFS” à la page 25
- “Amélioration de la sortie de la commande `zpool status`” à la page 26
- “Améliorations de ZFS et Solaris iSCSI” à la page 27

- “Historique de commande ZFS (zpool history)” à la page 27
- “Améliorations de propriétés ZFS” à la page 28
- “Affichage de la totalité des informations de systèmes de fichiers ZFS” à la page 29
- “Nouvelle option zfs receive -F” à la page 29
- “Instantanés ZFS récursifs” à la page 30
- “Double Parité RAID-Z (raidz2)” à la page 30
- “Disques hot spare pour périphériques de pool de stockage ZFS” à la page 30
- “Remplacement d'un système de fichiers ZFS par un clone ZFS (zfs promote)” à la page 31
- “Mise à niveau des pools de stockage ZFS (zpool upgrade)” à la page 31
- “Commandes de sauvegarde et de rétablissement ZFS renommées” à la page 31
- “Récupération de pools de stockage détruits” à la page 31
- “Intégration de ZFS au gestionnaire de pannes” à la page 32
- “Nouvelle commande zpool clear” à la page 32
- “Format NFSv4 ACL compact” à la page 32
- “Outil de contrôle de système de fichiers (fsstat)” à la page 33
- “Gestion Web ZFS” à la page 33

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

Version Solaris 10 10/08 : cette version permet d'installer et d'initialiser un système de fichiers racine ZFS. Vous pouvez utiliser l'option d'installation initiale ou la fonction JumpStart pour installer un système de fichiers racine ZFS. Vous pouvez sinon utiliser la fonction Live Upgrade pour migrer d'un système de fichiers racine UFS vers un système de fichiers racine ZFS. La prise en charge ZFS des périphériques de swap et de vidage est également disponible. Pour plus d'informations, reportez-vous au [Chapitre4](#), “Installation et initialisation d'un système de fichiers racine ZFS”.

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

Rétablissement d'un jeu de données sans démontage

Version Solaris 10 10/08 : cette version permet de rétablir un jeu de données sans avoir à le démonter au préalable. Cette fonction signifie que l'option `zfs rollback -f` n'est plus nécessaire pour effectuer un démontage forcé. L'option `-f` n'est plus prise en charge et est ignorée si elle est spécifiée.

Améliorations apportées à la commande `zfs send`

Version Solaris 10 10/08 D : les améliorations suivantes ont été apportées à la commande `zfs send` dans cette version.

- Envoi de tous les flux incrémentiels d'un instantané vers un instantané cumulatif. Exemple :

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

Tous les instantanés incrémentiels de `fs@snapA` à `fs@snapD` sont envoyés vers `fs@combo`.

- Envoi d'un flux incrémentiel à partir 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. Exemple :

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

- Envoi d'un flux de réplication de tous les systèmes de fichiers descendants, jusqu'aux instantanés nommés. Une fois reçus, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés. Exemple :

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

Pour une illustration détaillée, reportez-vous à l'[Exemple 7-1](#).

- Envoi d'un flux de réplication incrémentiel.

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

Pour une illustration détaillée, reportez-vous à l'[Exemple 7-1](#).

Pour plus d'informations, reportez-vous à la section “[Envoi et réception de flux d'instantanés ZFS complexes](#)” à la page 187.

Quotas et réservations ZFS pour les données de système de fichiers uniquement

Version Solaris 10 10/08 : outre les fonctionnalités de quotas et de réservation ZFS, cette version fournit des quotas et réservations de jeux de données excluant les données descendantes, telles que les instantanés et les clones, lors de la comptabilisation de l'utilisation de l'espace.

- La propriété `refquota` limite la quantité d'espace consommable par un jeu de données. Cette propriété définit une quantité d'espace utilisable maximale. Cette limite fixe n'inclut pas l'espace utilisé par les descendants, tels que les instantanés et les clones.
- La propriété `refreservation` définit la quantité minimale d'espace accordée à un jeu de données (descendants exclus).

Par exemple, en définissant la propriété `refquota` de `studentA` sur 10 Go, vous spécifiez une limite fixe de 10 Go d'espace *référéncé*. Pour une plus grande flexibilité, vous pouvez définir un quota de 20 Go qui vous permet de gérer les instantanés de `studentA`.

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

Pour plus d'informations, reportez-vous à la section [“Quotas et réservations ZFS” à la page 170](#).

Propriétés de pool de stockage ZFS

Version Solaris 10 10/08 : les propriétés de pool de stockage ZFS ont été présentées dans une version antérieure. Cette version fournit des informations sur les propriétés supplémentaires. Exemple :

```
# zpool get all mpool
NAME PROPERTY VALUE SOURCE
mpool size 33.8G -
mpool used 5.76G -
mpool available 28.0G -
mpool capacity 17% -
mpool altroot - default
mpool health ONLINE -
mpool guid 2689713858991441653 -
mpool version 10 default
mpool bootfs mpool/ROOT/zfsBE local
mpool delegation on default
mpool autoreplace off default
mpool cachefile - default
mpool failmode continue local
```

Pour une description de ces propriétés, reportez-vous au [Tableau 5–1](#).

- Propriété `cachefile` – **Version Solaris 10 10/08** : cette version fournit la propriété `cachefile` qui contrôle l'emplacement de mise en cache des informations de configuration de pool. Tous les pools du cache sont importés automatiquement au démarrage 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 afin de mettre la configuration de pool en cache à un autre emplacement. Il reste alors possible d'importer ultérieurement ce dernier à l'aide de la commande `zpool import c`. Cette propriété n'est pas utilisée dans la plupart des configurations ZFS.

La propriété `cachefile` n'est pas persistante et n'est pas stockée sur le disque. Elle remplace la propriété `temporary` qui, dans les versions précédentes de Solaris, indiquait que les informations de pool ne devaient pas être mises en cache.

- Propriété `failmode` – **Version Solaris 10 10/08** : dans cette version, la propriété `failmode` permet de déterminer l'action à effectuer en cas d'échec catastrophique de pool causé par la perte de connectivité d'un périphérique ou la panne de tous les périphériques du pool. Vous pouvez définir la propriété `failmode` sur l'une des valeurs suivantes : `wait`, `continue` ou `panic`. La valeur par défaut est `wait` : vous devez reconnecter le périphérique ou remplacer le périphérique défaillant, puis effacer l'erreur à l'aide de la commande `zpool clear`.

La propriété `failmode` est définie comme les autres propriétés ZFS définissables, avant ou après la création du pool. Exemple :

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

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

Pour une description des propriétés de pool ZFS, reportez-vous au [Tableau 5-1](#).

Améliorations apportées à l'historique des commandes ZFS (`zpool history`)

Version Solaris 10 10/08 : la commande `zpool history` a été améliorée afin de fournir les nouvelles fonctions suivantes :

- Les informations sur les événements des systèmes de fichiers ZFS s'affichent. Exemple :

```
# zpool history users
History for 'users':
2008-07-10.09:43:05 zpool create users mirror c1t1d0 c1t2d0
2008-07-10.09:43:48 zfs create users/home
```

```
2008-07-10.09:43:56 zfs create users/home/markm
2008-07-10.09:44:02 zfs create users/home/marks
2008-07-10.09:44:19 zfs snapshot -r users/home@yesterday
```

- L'option `-l` permet d'afficher un format complet comprenant le nom de l'utilisateur, le nom de l'hôte et la zone dans laquelle l'opération a été effectuée. Exemple :

```
# zpool history -l users
```

```
History for 'users':
```

```
2008-07-10.09:43:05 zpool create users mirror c1t1d0 c1t2d0 [user root on corona:global]
2008-07-10.09:43:13 zfs create users/marks [user root on corona:global]
2008-07-10.09:43:44 zfs destroy users/marks [user root on corona:global]
2008-07-10.09:43:48 zfs create users/home [user root on corona:global]
2008-07-10.09:43:56 zfs create users/home/markm [user root on corona:global]
2008-07-10.09:44:02 zfs create users/home/marks [user root on corona:global]
2008-07-11.10:44:19 zfs snapshot -r users/home@yesterday [user root on corona:global]
```

- Option `-i` pour l'affichage des informations relatives aux événements internes utilisables pour établir des diagnostics. Exemple :

```
# zpool history -i users
```

```
History for 'users':
```

```
2008-07-10.09:43:05 zpool create users mirror c1t1d0 c1t2d0
2008-07-10.09:43:13 [internal create txg:6] dataset = 21
2008-07-10.09:43:13 zfs create users/marks
2008-07-10.09:43:48 [internal create txg:12] dataset = 27
2008-07-10.09:43:48 zfs create users/home
2008-07-10.09:43:55 [internal create txg:14] dataset = 33
2008-07-10.09:43:56 zfs create users/home/markm
2008-07-10.09:44:02 [internal create txg:16] dataset = 39
2008-07-10.09:44:02 zfs create users/home/marks
2008-07-10.09:44:19 [internal snapshot txg:21] dataset = 42
2008-07-10.09:44:19 [internal snapshot txg:21] dataset = 44
2008-07-10.09:44:19 [internal snapshot txg:21] dataset = 46
2008-07-10.09:44:19 zfs snapshot -r users/home@yesterday
```

Pour plus d'informations sur l'utilisation de la commande `zpool history`, reportez-vous à la section [“Identification de problèmes dans ZFS”](#) à la page 251.

Mise à niveau des systèmes de fichiers ZFS (zfs upgrade)

Version Solaris 10 10/08 : la commande `zfs upgrade` a été ajoutée à cette version pour pouvoir apporter des améliorations futures aux systèmes de fichiers ZFS existants. Les pools de stockage disposent d'une fonctionnalité de mise à niveau similaire permettant d'apporter des améliorations aux pools de stockage existants.

Exemple :

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

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

Remarque – Les systèmes de fichiers mis à niveau et tout flux créé à partir de ces systèmes de fichiers mis à niveau à l'aide de la commande `zfs send` ne sont pas accessibles sur des systèmes exécutant des versions antérieures du logiciel.

Administration déléguée de ZFS

Version Solaris 10 10/08 : cette version vous permet de déléguer des droits précis à des utilisateurs sans privilèges pour qu'ils effectuent des tâches d'administration ZFS.

Les commandes `zfs allow` et `zfs unallow` permettent d'accorder ou de retirer les autorisations.

Vous pouvez modifier la capacité d'utilisation de l'administration déléguée à l'aide de la propriété `delegation` du pool. 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.

Pour plus d'informations, reportez-vous au [Chapitre9, “Administration déléguée de ZFS”](#) et à la page de manuel [zfs\(1M\)](#).

Configuration de périphériques de journalisation ZFS distincts

Version Solaris 10 10/08 : 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 assurer la stabilité des données à l'aide de `fsync()`. Par défaut, le ZIL est attribué à

partir de blocs dans le pool de stockage principal. Cependant, les performances peuvent parfois être améliorées en vous servant de périphériques de journalisation d'intention distincts dans votre pool de stockage ZFS, notamment d'une NVRAM ou d'un disque dédié.

Les périphériques de journalisation du ZIL ne sont pas liés aux fichiers journaux de base de données.

Vous pouvez configurer un périphérique de journalisation ZFS à la création du pool de stockage ou à un moment ultérieur. Pour obtenir des exemples de configuration de périphériques de journalisation, reportez-vous à la section “Création d'un pool de stockage ZFS avec des périphériques de journalisation” à la page 99 et à la section “Ajout de périphériques à un pool de stockage” à la page 106.

connexion d'un périphérique de journal à un périphérique journal 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.

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

- 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.sun.com/perrin/entry/slog_blog_or_blogging_on
- Les périphériques de journalisation peuvent être mis en miroir et leur réplication peut être annulée, mais RAIDZ 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. Les périphériques de journalisation ne peuvent pour le moment pas être supprimés.
- 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.

Création de jeux de données ZFS intermédiaires

Version Solaris 10 10/08 : vous pouvez appliquer l'option `-p` aux commandes `zfs create`, `zfs clone` et `zfs rename` afin de créer rapidement un jeu de données intermédiaire s'il n'existe pas encore.

Par exemple, créez les jeux de données ZFS `users/area51` dans le pool de stockage `datab`.

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

Si le jeu de données intermédiaire existe pendant l'opération de création, celle-ci est réussie.

Les propriétés spécifiées s'appliquent au jeu de données cible, mais pas aux jeux de données intermédiaires. Exemple :

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

Le jeu de données intermédiaire est créé avec le point de montage par défaut. Toute propriété supplémentaire est désactivée pour ce jeu de données. Exemple :

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

Pour de plus amples informations, reportez-vous à la page de manuel [zfs\(1M\)](#).

Améliorations apportées à la connexion à chaud à ZFS

Version Solaris 10 10/08 : dans cette version, ZFS répond de manière plus efficace aux périphériques supprimés et fournit un mécanisme permettant d'identifier automatiquement des périphériques insérés, avec les améliorations suivantes :

- Vous pouvez remplacer un périphérique existant par un périphérique équivalent sans utiliser la commande `zpool replace`.

La propriété `autoreplace` contrôle le remplacement automatique de périphériques. Si la propriété est désactivée, l'administrateur doit initier le remplacement du périphérique à l'aide de la commande `zpool replace`. Si la propriété est activée, tout nouveau périphérique se trouvant au même emplacement physique qu'un périphérique qui appartenait au pool est automatiquement formaté et remplacé. Le comportement par défaut est "off".

- L'état de pool de stockage `REMOVED` est fourni lors du retrait du périphérique ou d'un disque hot spare, en cas de retrait physique du périphérique 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 inséré, 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.

Pour de plus amples informations, reportez-vous à la page de manuel [zpool\(1M\)](#).

Renommage récursif d'instantanés ZFS (`zfs rename -r`)

Version Solaris 10 10/08 : vous pouvez renommer tous les instantanés ZFS descendants de manière récursive à l'aide de la commande `zfs rename -r`.

Prenez par exemple un instantané de plusieurs systèmes de fichiers ZFS.

```
# zfs snapshot -r users/home@today
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPPOINT
users	216K	16.5G	20K	/users
users/home	76K	16.5G	22K	/users/home
users/home@today	0	-	22K	-
users/home/markm	18K	16.5G	18K	/users/home/markm
users/home/markm@today	0	-	18K	-
users/home/marks	18K	16.5G	18K	/users/home/marks
users/home/marks@today	0	-	18K	-
users/home/neil	18K	16.5G	18K	/users/home/neil
users/home/neil@today	0	-	18K	-

Renommez ensuite les instantanés de la manière suivante.

```
# zfs rename -r users/home@today @yesterday
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPPOINT
users	216K	16.5G	20K	/users
users/home	76K	16.5G	22K	/users/home

users/home@yesterday	0	-	22K	-
users/home/markm	18K	16.5G	18K	/users/home/markm
users/home/markm@yesterday	0	-	18K	-
users/home/marks	18K	16.5G	18K	/users/home/marks
users/home/marks@yesterday	0	-	18K	-
users/home/neil	18K	16.5G	18K	/users/home/neil
users/home/neil@yesterday	0	-	18K	-

Les instantanés sont le seul jeu de données qu'il est possible de renommer de façon récursive.

Pour plus d'informations sur les instantanés, reportez-vous à la section “[Présentation des instantanés ZFS](#)” à la page 177 et à l'entrée de blog décrivant la création d'instantanés restaurés :

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

La compression GZIP est disponible pour ZFS

Versión Solaris 10 10/08 : dans cette version de Solaris, vous pouvez définir la compression `gzip` sur des systèmes de fichiers ZFS en plus de la compression `lzjb`. Vous pouvez spécifier la compression `gzip` (format par défaut) ou `gzip-N` où `N` correspond à un chiffre entre 1 et 9. Exemple :

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

Pour plus d'informations sur la configuration des propriétés ZFS, reportez-vous à la section “[Définition des propriétés ZFS](#)” à la page 158.

Stockage de plusieurs copies de données utilisateur ZFS

Versión Solaris 10 10/08 : à des fins de fiabilité, les métadonnées d'un système de fichiers ZFS sont automatiquement stockées plusieurs fois sur divers disques, si possible. Cette fonction est connue sous le terme anglais de *ditto blocks*.

Cette version vous permet également de demander à stocker plusieurs copies des données utilisateur par système de fichiers à l'aide de la commande `zfs set copies`. Exemple :

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

Les valeurs disponibles sont 1, 2 et 3. La valeur par défaut est 1. Ces copies constituent un ajout à toute redondance de niveau pool, par exemple dans une configuration en miroir ou RAID-Z.

Stocker plusieurs copies des données utilisateur ZFS présente les avantages suivants :

- Cela améliore la rétention des données en autorisant leur récupération à partir d'erreurs de lecture de blocs irrécupérables, comme par exemple des défaillances de média pour l'ensemble des configurations ZFS.
- Cela garantit la sécurité des données même si un seul disque est disponible.
- Cela permet de choisir les stratégies de protection des données par système de fichiers et de dépasser les capacités du pool de stockage.

Selon l'allocation des blocs "ditto" dans le pool de stockage, plusieurs copies peuvent être placées sur un seul disque. La saturation ultérieure d'un disque peut engendrer l'indisponibilité de tous les blocs "ditto".

Vous pouvez envisager l'utilisation des blocs "ditto" lorsque vous créez accidentellement un pool non redondant et lorsque vous avez besoin de définir des stratégies de conservation de données.

Pour une description détaillée de l'impact de la configuration de copies sur un système comprenant un pool d'un seul disque ou un pool de plusieurs disques, sur la protection globale des données, consultez le blog suivant :

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

Pour plus d'informations sur la configuration des propriétés ZFS, reportez-vous à la section "Définition des propriétés ZFS" à la page 158.

Amélioration de la sortie de la commande `zpool status`

Version Solaris 10 8/07 : la commande `zpool status -v` permet d'afficher la liste des fichiers comportant des erreurs persistantes. Auparavant, il fallait utiliser la commande `find -inum` pour identifier les noms de fichiers à partir de la liste d'inodes affichée.

Pour obtenir des informations supplémentaires sur l'affichage d'une liste de fichiers comportant des erreurs persistantes, consultez la section "Réparation d'un fichier ou répertoire corrompu" à la page 269.

Améliorations de ZFS et Solaris iSCSI

Version Solaris 10 8/07 : dans cette version de Solaris, vous pouvez créer un volume ZFS en tant que périphérique cible Solaris iSCSI en configurant la propriété `shareiscsi` sur le volume ZFS. Cette méthode permet de configurer rapidement et facilement une cible Solaris iSCSI.

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 obtenir des informations sur la configuration d'un initiateur Solaris iSCSI, consultez le [Chapitre 14, “Configuring Solaris iSCSI Targets and Initiators \(Tasks\)”](#) du *System Administration Guide: Devices and File Systems*.

Pour obtenir des informations supplémentaires sur la gestion d'un volume ZFS en tant que cible iSCSI, consultez la section [“Utilisation d'un volume ZFS en tant que cible iSCSI Solaris”](#) à la page 237.

Historique de commande ZFS (zpool history)

Version Solaris 10 8/07 : dans cette version de Solaris, ZFS consigne automatiquement les commandes `zfs` et `zpool` qui modifient les informations concernant l'état du pool. Exemple :

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

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

Cette fonction vous permet, ainsi qu'au personnel de support Sun, d'identifier le jeu *exact* de commandes ZFS exécutées pour résoudre un scénario d'erreur.

Vous pouvez identifier un pool de stockage spécifique grâce à la commande `zpool history`. Exemple :

```
# zpool history newpool
History for 'newpool':
2007-04-25.11:37:31 zpool create newpool mirror c0t8d0 c0t10d0
```

```
2007-04-25.11:37:46 zpool replace newpool c0t10d0 c0t9d0
2007-04-25.11:38:04 zpool attach newpool c0t9d0 c0t11d0
2007-04-25.11:38:09 zfs create newpool/user1
2007-04-25.11:38:15 zfs destroy newpool/user1
```

Les caractéristiques de l'historique sont les suivantes :

- Le journal ne peut pas être désactivé.
- Le journal est enregistré sur le disque de façon persistante, ce qui signifie qu'il est enregistré au travers des réinitialisations du système.
- 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 maximum est plafonnées à 1 % de la taille du pool, la *taille* étant déterminée lors de la création du pool.
- Ne nécessite aucune administration, ce qui signifie qu'il n'est pas nécessaire d'ajuster la taille du journal ou de modifier son emplacement.

Dans cette version de Solaris, la commande `zpool history` n'enregistre pas l'*ID utilisateur*, le *nom d'hôte* ni le *nom de la zone*. Pour plus d'informations, reportez-vous à la section [“Améliorations apportées à l'historique des commandes ZFS \(`zpool history`\)”](#) à la page 19.

Pour obtenir des informations supplémentaires sur la résolution de problèmes relatifs à ZFS, consultez la section [“Identification de problèmes dans ZFS”](#) à la page 251.

Améliorations de propriétés ZFS

Propriété ZFS `xattr`

Versión Solaris 10 8/07 : vous pouvez utiliser la propriété `xattr` pour désactiver ou activer les attributs avancés d'un système de fichiers ZFS spécifique. Par défaut, les attributs sont activés. Pour obtenir une description des propriétés ZFS, reportez-vous à la section [“Présentation des propriétés ZFS”](#) à la page 143.

Propriété ZFS `canmount`

Versión Solaris 10 8/07 : la nouvelle propriété `canmount` permet de spécifier s'il est possible de monter un jeu de données à l'aide de la commande `zfs mount`. Pour plus d'informations, reportez-vous à la section [“Propriété `canmount`”](#) à la page 153.

Propriétés ZFS définies par l'utilisateur

Versión Solaris 10 8/07 : outre les propriétés natives standard qui permettent d'exporter des statistiques internes ou de contrôler le comportement du système de fichiers ZFS, ZFS assure la prise 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.

Pour obtenir des informations supplémentaires, consultez la section [“Propriétés ZFS définies par l'utilisateur”](#) à la page 154.

Définition des propriétés lors de la création de systèmes de fichiers ZFS

Version Solaris 10 8/07 : dans cette version de Solaris, vous pouvez non seulement définir les propriétés d'un système de fichiers existant, mais vous avez également la possibilité de définir les propriétés lors de la création du système de fichiers.

Les exemples suivants montrent la syntaxe équivalente :

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

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

Affichage de la totalité des informations de systèmes de fichiers ZFS

Version Solaris 10 8/07 : dans cette version de Solaris, vous pouvez utiliser diverses formes de la commande `zfs get` pour afficher des informations sur l'ensemble des jeux de données si vous n'en spécifiez pas un en particulier ou si vous ne spécifiez pas `all`. Dans les versions précédentes, la commande `zfs get` ne permettait pas de récupérer la totalité des informations relatives au jeu de données.

Exemple :

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

Nouvelle option `zfs receive -F`

Solaris 10 8/07 : dans cette version de Solaris, vous pouvez utiliser la nouvelle option `-F` avec la commande `zfs receive` pour forcer une restauration du système de fichiers à l'instantané le plus récent avant d'effectuer la réception. Cette option peut s'avérer nécessaire en cas de modification du système entre le moment où une restauration se produit et celui où la réception est initialisée.

Pour plus d'informations, reportez-vous à la section [“Réception d'un instantané ZFS”](#) à la page 186.

Instantanés ZFS récursifs

Versión Solaris 10 11/06 : lorsque vous créez un instantané de système de fichiers à l'aide de la commande `zfs snapshot`, spécifiez l'option `-r` si vous souhaitez créer des instantanés de façon récursive pour l'ensemble des systèmes de fichiers descendants. Cette option `-r` permet en outre de détruire récursivement tous les instantanés descendants lors de la destruction d'un instantané.

Une seule opération, dite atomique, permet de créer rapidement des instantanés ZFS récursifs. Ceux-ci sont tous créés simultanément ou ne sont pas créés du tout. Grâce à ce type de sélection instantanée, les opérations atomiques assurent ainsi la cohérence des données, y compris pour les systèmes de fichiers descendants.

Pour obtenir des informations supplémentaires, consultez la section [“Création et destruction d'instantanés ZFS”](#) à la page 178.

Double Parité RAID-Z (`raidz2`)

Versión Solaris 10 11/06 : une configuration RAID-Z redondante peut désormais présenter une parité simple ou double. En d'autres termes, le système peut subir respectivement une ou deux pannes de périphérique sans perte de données. Le mot-clé `raidz2` permet de spécifier une configuration RAID-Z à deux parités. Pour spécifier une configuration RAID-Z à une parité, vous avez le choix entre les mots-clés `raidz` et `raidz1`.

Pour plus d'informations, consultez la section [“Création de pools de stockage RAID-Z”](#) à la page 98 ou la page de manuel `zpool(1M)`.

Disques hot spare pour périphériques de pool de stockage ZFS

Versión Solaris 10 11/06 : La fonction de disque hot spare permet d'identifier les disques pouvant être utilisés pour remplacer un périphérique défaillant dans un ou plusieurs pools de stockage. Le périphérique *hot spare* peut immédiatement remplacer tout périphérique actif du pool qui tombe en panne. Vous pouvez également effectuer ce remplacement manuellement.

Pour plus d'informations, reportez-vous à la section [“Désignation des disques hot spare dans le pool de stockage”](#) à la page 116 et à la page de manuel `zpool(1M)`.

Remplacement d'un système de fichiers ZFS par un clone ZFS (`zfs promote`)

Version Solaris 10 11/06 : la commande `zfs promote` permet de remplacer un système de fichiers ZFS existant par un clone de celui-ci. Cette fonctionnalité permet de tester la nouvelle version d'un système de fichiers, puis de la définir comme système de fichiers actif.

Pour plus d'informations, reportez-vous à la section [“Remplacement d'un système de fichiers ZFS par un clone ZFS”](#) à la page 183 et à la page de manuel `zfs(1M)`.

Mise à niveau des pools de stockage ZFS (`zpool upgrade`)

Version Solaris 10 6/06 : vous pouvez mettre à niveau vos pools de stockage à l'aide de la commande `zpool upgrade` et tirer ainsi profit des fonctions les plus récentes. De plus, la commande `zpool status` indique dorénavant si la version actuelle des pools est plus ancienne.

Pour plus d'informations, reportez-vous à la section [“Mise à niveau de pools de stockage ZFS”](#) à la page 137 et à la page de manuel `zpool(1M)`.

Si vous souhaitez utiliser la console d'administration ZFS dans un système comprenant un pool provenant d'une version précédente de Solaris, veuillez préalablement à mettre les pools à niveau. La commande `zpool status` permet de savoir si une mise à niveau des pools est requise. Pour obtenir des informations sur la console d'administration ZFS, reportez-vous à la rubrique [“Gestion Web ZFS”](#) à la page 33.

Commandes de sauvegarde et de rétablissement ZFS renommées

Version Solaris 10 6/06 : dans cette version de Solaris les commandes `zfs backup` et `zfs restore` ont été renommées `zfs send` et `zfs receive` afin de mieux décrire leur fonction. Ces commandes ont pour fonction d'enregistrer et de restaurer les représentations de flux de données ZFS.

Pour plus d'informations sur ces commandes, reportez-vous à la section [“Envoi et réception de données ZFS”](#) à la page 184.

Récupération de pools de stockage détruits

Version Solaris 10 6/06 : cette version inclut la commande `zpool import -D` qui permet la récupération de pools précédemment détruits à l'aide de la commande `zpool destroy`.

Pour obtenir des informations supplémentaires, reportez-vous à la section “[Récupération de pools de stockage ZFS détruits](#)” à la page 135.

Intégration de ZFS au gestionnaire de pannes

Version Solaris 10 6/06 : cette version intègre un moteur de diagnostic ZFS capable de diagnostiquer et de signaler les pannes des périphériques et des pools. Les erreurs liées aux sommes de contrôle, aux E/S et aux périphériques font également l'objet d'un rapport lorsqu'elles sont liées à la défaillance d'un pool ou d'un périphérique.

Le moteur de diagnostic n'effectue pas d'analyses prédictives d'erreurs liées aux sommes de contrôle ou aux E/S et n'inclut aucune action proactive basée sur l'analyse de pannes.

En cas de panne ZFS, la commande `fmd` peut émettre un message tel que le suivant :

```
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Tue Mar 18 21:48:06 MDT 2008
PLATFORM: SUNW,Ultra-Enterprise, CSN: -, HOSTNAME: neo
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: f1ae0cad-f2dd-cfdc-a821-a3be5b363d68
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.
```

Il est recommandé de suivre les instructions spécifiques de la commande `zpool status`, afin d'identifier et de résoudre rapidement la panne.

Pour obtenir un exemple de récupération à partir d'un problème ZFS signalé, reportez-vous à la section “[Réparation d'un périphérique manquant](#)” à la page 257.

Nouvelle commande `zpool clear`

Version Solaris 10 6/06 : cette version inclut la commande `zpool clear` permettant d'effacer les décomptes d'erreurs associés à un périphérique ou au pool. Auparavant, les décomptes d'erreurs étaient effacés lors de la mise en ligne du périphérique d'un pool avec la commande `zpool online`. Pour plus d'informations, reportez-vous à la page de manuel `zpool(1M)` et à la section “[Suppression des périphériques de pool de stockage](#)” à la page 114.

Format NFSv4 ACL compact

Version Solaris 10 6/06 : cette version offre trois formats ACL NFSv4 : détaillé, positionnel et compact. Les nouveaux formats ACL compact et positionnel permettent de définir et d'afficher

les listes de contrôle. La commande `chmod` permet de définir les trois formats ACL. Vous pouvez utiliser la commande `ls -V` pour l'affichage des formats ACL compact et positionnel et la commande `ls -v` pour l'affichage du format ACL détaillé.

Pour plus d'informations, reportez-vous à la section “[Configuration et affichage d'ACL dans des fichiers ZFS en format compact](#)” à la page 216, à la page de manuel `chmod(1)` et à la page de manuel `ls(1)`.

Outil de contrôle de système de fichiers (`fsstat`)

Versión Solaris 10 6/06 : un nouvel outil de contrôle de système de fichiers, `fsstat`, est disponible pour la génération de rapports relatifs aux opérations de système de fichiers. 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 zfs
new name name attr attr lookup rmdir read read write write
file remov chng get set ops ops ops bytes ops bytes
7.82M 5.92M 2.76M 1.02G 3.32M 5.60G 87.0M 363M 1.86T 20.9M 251G zfs
```

Pour de plus amples informations, reportez-vous à la page de manuel `fsstat(1M)`.

Gestion Web ZFS

Versión Solaris 10 6/06 : un outil Web de gestion ZFS permet d'effectuer de nombreuses opérations administratives. Cet outil permet de réaliser les tâches 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 ;
- créer un volume ;
- prendre 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 via un navigateur Web sécurisé à l'URL suivant :

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

Si vous avez saisi l'URL correct 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 exécuter 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 la console de gestion Solaris (smc, Solaris Management Console).

Description de ZFS

Le système de fichiers ZFS présente des fonctions et des avantages uniques au monde. Ce système de fichiers révolutionnaire modifie radicalement les méthodes d'administration des systèmes de fichiers. ZFS a été conçu afin d'être robuste, évolutif et facile à gérer.

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 l'image 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.

Grâce à ZFS, la gestion de volumes devient inutile. 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 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 augmentent automatiquement au sein de l'espace 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 supplémentaire, sans requérir des tâches supplémentaires. Selon divers aspects, le pool de stockage agit comme un système de mémoire virtuel. Lors de

l'ajout d'un DIMM mémoire à un système, le système d'exploitation ne vous force pas à configurer la mémoire et à l'assigner à des processus, à l'aide de commandes spécifiques. 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 de la machine, 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. Elle parcourait et vérifiait l'état du système de fichiers et tentait de réparer les incohérences détectées. Cette situation représentait une grande source de problème pour les administrateurs et la résolution de tous les problèmes potentiels n'était pas garantie. 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 sont 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. Grâce à ce mécanisme, la corruption du système de fichier en raison d'une coupure de courant ou d'un arrêt du système est impossible. Ainsi, aucune commande n'est requise pour remplacer `fsck`. Il est possible que les données récemment écrites soient perdues, mais le système de fichiers reste toujours 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, une somme de contrôle des données et métadonnées est effectué selon l'algorithme 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. Dans la conception classique, certains modes d'échec, comme l'écriture d'un bloc complet dans un emplacement incorrect, peut générer des données dont la somme de contrôle s'est effectuée correctement mais qui sont en réalité incorrectes. Les somme de contrôle ZFS sont stockées de manière à ce que ces modes d'échec soient détectés et qu'une solution y soit apportée. Toutes les opérations de contrôle de somme et de récupération des données sont effectuées sur la couche système de fichiers et sont transparentes aux applications.

De plus, ZFS fournit des données d'autorétablissement. ZFS prend en charge les pools de stockage avec divers niveaux de redondance de données, y compris la mise en miroir et une variante de RAID-5. 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.

Évolutivité inégalée

ZFS a été conçu dès le départ avec l'objectif d'en faire le système de fichiers le plus évolutif ayant jamais existé. La taille du système de fichiers lui-même est de 128 bits et vous pouvez utiliser jusqu'à 256 quadrillion 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 supplémentaire dans le pool.

À mesure que le jeu de données actif est modifié, l'espace 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 examiner ou réparer les périphériques sans utiliser un jeu distinct de commandes de gestion de volumes. Vous pouvez prendre un nombre illimité d'instantanés de systèmes de fichiers. Vous pouvez sauvegarder et restaurer des systèmes de fichiers individuels.

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 deviennent le point de contrôle central. Les systèmes de

fichiers eux-mêmes étant très peu coûteux (autant qu'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 à l'aide de la commande <code>lucreate</code> et éventuellement mis à jour à l'aide de la commande <code>luupgrade</code> mais qui n'est pas actuellement l'environnement d'initialisation principal ou actif. L'environnement d'initialisation alternatif peut être remplacé par l'environnement d'initialisation principal en exécutant la commande <code>luactivate</code> .
Somme de contrôle	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 <code>fletcher2</code> (paramètre par défaut), ainsi que les puissantes fonctions de hachage cryptographique telles que <code>SHA256</code> .
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 182.
Jeu de données	Nom générique pour les entités ZFS suivantes : clones, systèmes de fichiers, instantanés ou 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 : <i>pool/chemin</i> [<i>@instantané</i>] <i>pool</i> Identifie le nom d'un pool de stockage contenant le jeu de données. <i>chemin</i> Nom de chemin délimité par slash pour l'objet du jeu de données.

	<p><i>instantané</i> Composant optionnel identifiant l'instantané d'un jeu de données.</p> <p>Pour plus d'informations sur les jeux de données, reportez-vous au Chapitre6, "Gestion des systèmes de fichiers ZFS".</p>
Systèmes de fichiers par défaut	<p>Systèmes de fichiers créés par défaut lors de l'utilisation de Live upgrade pour migrer d'un système racine UFS à un système racine ZFS. Le groupe actuel de systèmes de fichiers par défaut est le suivant :</p> <pre> / /usr /opt /var </pre>
Système de fichiers	<p>Jeu de données ZFS de type <code>filesystem</code> monté au sein de l'espace de noms système standard et se comportant comme les autres systèmes de fichiers.</p> <p>Pour plus d'informations sur les systèmes de fichiers, reportez-vous au Chapitre6, "Gestion des systèmes de fichiers ZFS".</p>
Miroir	<p>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.</p>
Pool	<p>Groupe logique de périphériques décrivant la disposition et les caractéristiques physiques du stockage disponible. L'espace pour les jeux de données est alloué à partir d'un pool.</p> <p>Pour plus d'informations sur les pools de stockage, reportez-vous au Chapitre5, "Gestion des pools de stockage ZFS".</p>
Environnement d'initialisation principal	<p>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</p>

	actuel. Ce paramètre par défaut peut être modifié à l'aide de l'option <code>lucreate - s</code> .
RAID-Z	Périphérique virtuel stockant les données et la parité sur plusieurs disques, similaire à RAID-5. Pour plus d'informations sur RAID-Z, reportez-vous à la section " Configuration de pool de stockage RAID-Z " à la page 94.
Réargenture	Processus de transfert de données d'un périphérique à un autre. Par exemple, si un composant de miroir est remplacé ou mis hors ligne, les données du composant de miroir le plus actuel est copié dans le composant de miroir nouvellement restauré. Dans les produits de gestion de volumes classiques, ce processus est appelé <i>resynchronisation de miroir</i> . Pour plus d'informations sur la réargenture ZFS, reportez-vous à la section " Affichage de l'état de réargenture " à la page 265.
Systèmes de fichiers partagés	Groupe de systèmes de fichiers partagés entre l'environnement d'initialisation alternatif et l'environnement d'initialisation principal. Ce groupe inclut des systèmes de fichiers, notamment <code>/export</code> , ainsi que la zone réservée au swap. Les systèmes de fichiers partagés peuvent également contenir des racines de zone.
Instantané	Image en lecture seule d'un système de fichiers ou d'un volume à un instant <code>t</code> . Pour plus d'informations sur les instantanés, reportez-vous à la section " Présentation des instantanés ZFS " à la page 177.
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 100.

Volume

Jeu de données utilisé pour émuler un périphérique physique. 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 235.

Exigences d'attribution de noms de composants ZFS

L'attribution de noms de chaque composant ZFS doit respecter les règles suivantes :

- Les composants vides sont interdits.
- 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 pools doivent commencer par une lettre, à l'exception des restrictions suivantes :
 - La séquence de début `c[0-9]` n'est pas autorisée
 - Le nom `log` est réservé
 - Les noms commençant par `mirror`, `raidz` ou `spare` ne sont pas autorisés, car ils sont réservés.

En outre, 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 (%).

Guide de démarrage de ZFS

Ce chapitre décrit les étapes à suivre pour définir une configuration ZFS simple. Il offre une vision globale du fonctionnement des commandes ZFS et explique les méthodes de création de pools et de systèmes de fichiers simples. Cette section ne constitue pas une présentation exhaustive. Pour des informations plus détaillées, reportez-vous aux autres chapitres, comme indiqué.

Il contient les sections suivantes :

- “Exigences et recommandations en matière de matériel et de logiciel ZFS” à la page 41
- “Création d'un système de fichiers ZFS basique” à la page 42
- “Création d'un pool de stockage ZFS” à la page 43
- “Création d'une hiérarchie de systèmes de fichiers ZFS” à la page 44

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 :

- Système SPARC™ ou x86 exécutant la version 10 6/06 ou une version plus récente de Solaris.
- La taille minimale du disque est de 128 Ko. L'espace disque minimum requis pour un pool de stockage est de 64 Mo.
- Il est actuellement recommandé de disposer d'au moins 768 Mo de mémoire pour installer un système Solaris. Cependant, pour assurer des performances correctes de ZFS, il est préférable de disposer d'au moins un gigaoctet de mémoire.
- En cas de création d'une configuration de disques mise en miroir, il est conseillé de disposer de plusieurs contrôleurs.

Création d'un système de fichiers ZFS basique

L'administration de ZFS a été conçue dans un but de simplicité. La conception de ZFS a notamment pour objectif de réduire le nombre de commandes nécessaires à la création d'un système de fichiers utilisable. 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 94.

Le nouveau système de fichiers ZFS, tank, peut utiliser autant d'espace de disque 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 souhaitez 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 de disque 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
```

Il s'avère souvent nécessaire de créer et d'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 44.

Création d'un pool de stockage ZFS

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.

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 44](#), partez du principe que les disques entiers `/dev/dsk/c1t0d0` et `/dev/dsk/c1t1d0` 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 91](#).

2 Sélectionnez la réplication de données.

ZFS assure la prise en charge de plusieurs types de réplication de données qui déterminent les types de pannes matérielles que pourra supporter 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).

L'exemple de stockage de la section [“Création d'un pool de stockage ZFS” à la page 44](#) 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 94](#).

▼ Création d'un pool de stockage ZFS

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

Pour de plus amples informations sur les droits de profils ZFS, reportez-vous à la section [“Profils de droits ZFS” à la page 245](#).

- 2 **Choisissez un nom de pool.**

Le nom de pool sert à identifier le pool de stockage lorsque vous exécutez les commandes `zpool` ou `zfs`. La plupart des systèmes ne requièrent qu'un pool. Vous pouvez donc utiliser le nom de votre choix tant qu'il respecte les exigences d'attribution de nom décrites dans la section [“Exigences d'attribution de noms de composants ZFS” à la page 40](#).

- 3 **Créez le pool.**

Par exemple, créez un pool mis en miroir nommé `tank`.

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

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

Pour de plus amples informations sur la création de pools de stockage, reportez-vous à la section [“Création d'un pool de stockage ZFS” à la page 97](#).

Pour plus d'informations sur la détection de l'utilisation de périphériques, reportez-vous à la section [“Détection des périphériques utilisés” à la page 102](#).

- 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  USED  AVAIL  CAP  HEALTH  ALROOT
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 123](#).

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 les systèmes de fichiers en hiérarchies arbitraires, dans lesquelles chaque système de fichiers n'a qu'un seul parent. La racine 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. Un modèle correct se compose d'un système de fichiers par utilisateur ou par projet. En effet, un tel modèle permet de contrôler les propriétés, les instantanés et les sauvegardes par utilisateur ou par projet.

Deux systèmes de fichiers ZFS, `bonwick` et `billm` sont créés dans la section [“Création de systèmes de fichiers ZFS”](#) à la page 46.

Pour plus d'informations sur la gestion des systèmes de fichiers, reportez-vous au [Chapitre 6](#), [“Gestion des systèmes de fichiers 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 46, 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 simples. 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 46, tous les répertoires de base sont montés dans `/export/zfs/utilisateur`. Ils sont partagés à l'aide de NFS et la compression est activée. De plus, un quota de 10 Go est appliqué dans `bonwick`.

Pour de plus amples informations sur les propriétés, reportez-vous à la section [“Présentation des propriétés ZFS”](#) à la page 143.

▼ Création de systèmes de fichiers ZFS

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

Pour de plus amples informations sur les droits de profils ZFS, reportez-vous à la section [“Profils de droits ZFS” à la page 245](#).

- 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
```

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

- 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

Une nouvelle fonction permettant de définir les propriétés de système de fichiers lors de la création du système de fichier est disponible. 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 143](#).

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

Les paramètres de propriétés de ces systèmes de fichiers sont hérités de leur parent. Ils sont donc montés sur `/export/zfs/ utilisateur` et partagés 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 140.

Pour de plus amples informations sur le montage et le partage de systèmes de fichiers, reportez-vous à la section [“Montage et partage des systèmes de fichiers ZFS”](#) à la page 164.

5 Définissez les propriétés spécifiques au système.

Dans cet exemple, un quota de 10 Go est attribué à l'utilisateur `bonwick`. 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/bonwick
```

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/billm      8K    67.0G   8K   /export/zfs/billm
tank/home/bonwick    8K    10.0G   8K   /export/zfs/bonwick
```

Notez que l'utilisateur `bonwick` ne dispose que de 10 Go d'espace alors que l'utilisateur `billm` peut utiliser l'intégralité 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 156.

Pour de plus amples informations sur l'utilisation et le calcul de l'espace, reportez-vous à la section [“Comptabilisation de l'espace ZFS”](#) à la page 50.

Différences entre ZFS et les systèmes de fichiers classiques

Cette section aborde les différences significatives entre ZFS et les systèmes de fichiers classiques. La compréhension de ces différences clé permet d'éviter les confusions lors de l'utilisation d'outils classiques en interaction avec ZFS.

Il contient les sections suivantes :

- [“Granularité du système de fichiers ZFS” à la page 49](#)
- [“Comptabilisation de l'espace ZFS” à la page 50](#)
- [“Comportement d'espace saturé” à la page 50](#)
- [“Montage de système de fichiers ZFS” à la page 51](#)
- [“Gestion de volumes classique” à la page 51](#)
- [“Nouveau modèle ACL Solaris” à la page 51](#)

Granularité du système de fichiers ZFS

Traditionnellement, les systèmes de fichiers étaient restreints à un périphérique, ce qui limitait les systèmes de fichiers eux-mêmes à la taille du 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 ont aidé 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 alloué au pool de stockage.

Au lieu de créer un système de fichier, comme `/export/home`, pour la gestion de plusieurs sous-répertoires d'utilisateurs, vous pouvez créer un système de fichiers par utilisateur. De plus, ZFS fournit une hiérarchie qui permet de paramétrer et de gérer facilement plusieurs systèmes de fichiers en appliquant des propriétés dont peuvent hériter les systèmes de fichiers de la hiérarchie.

Pour obtenir un exemple de création de hiérarchie de système de fichiers, reportez-vous à la section [“Création d'une hiérarchie de systèmes de fichiers ZFS” à la page 44](#).

Comptabilisation de l'espace ZFS

ZFS repose sur un concept de stockage mis en pool. 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 171. Les réservations permettent de garantir de l'espace disponible à un système de fichiers spécifique. Pour obtenir des informations sur les réservations, reportez-vous à la rubrique [“Définition de réservations sur les systèmes de fichiers ZFS”](#) à la page 173. 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, un coût d'espace immédiat est requis pour ces métadonnées lors de la création du système de fichiers. 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 fichier 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 utilisé disponible et référencé par un fichier ou un système de fichier 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 ne sont pas automatiquement mise à jour.

Comportement d'espace saturé

La création d'instantanés de systèmes de fichiers est peu coûteuse et facile dans ZFS. En général, les instantanés sont identiques dans la plupart des environnements ZFS. Pour plus d'informations sur les instantanés ZFS, reportez-vous au [Chapitre 7, “Utilisation des instantanés et des clones ZFS”](#).

La présence d'instantanés peut entraîner des comportements inattendus lors des tentatives de libération d'espace. En règle générale, si vous disposez des droits adéquats, vous pouvez supprimer un fichier d'un système de fichiers plein, ce qui entraîne une augmentation de la

quantité d'espace 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. 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, l'erreur ENOSPC ou EDQUOT peut se produire lorsque vous tentez de supprimer un fichier.

Montage de système de fichiers ZFS

ZFS a été conçu pour simplifier et faciliter l'administration. Par exemple, avec des systèmes de fichiers existants, 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 jeu de données. Vous n'avez pas besoin de gérer les entrées ZFS dans le fichier `/etc/vfstab`.

Pour de plus amples informations sur le montage et le partage de systèmes de fichiers ZFS, reportez-vous à la section [“Montage et partage des systèmes de fichiers ZFS”](#) à la page 164.

Gestion de volumes classique

Comme décrit à la section [“Stockage ZFS mis en pool”](#) à la page 34, 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.

Nouveau modèle ACL Solaris

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 :

- modèle basé sur la spécification NFSv4 et similaire aux ACL de type NT ;
- jeu de privilèges d'accès bien plus granulaire ;
- 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 sur l'utilisation de listes de contrôle d'accès (ACL) avec des fichiers ZFS, reportez-vous au [Chapitre 8](#), “[Utilisation des ACL pour la protection de fichiers ZFS](#)”.

Installation et initialisation d'un système de fichiers racine ZFS

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

Il contient les sections suivantes :

- “Installation et initialisation d'un système de fichiers racine ZFS (présentation)” à la page 53
- “Configuration requise pour l'installation de Solaris et de Solaris Live Upgrade pour la prise en charge de ZFS” à la page 55
- “Installation d'un système de fichiers racine ZFS (installation initiale)” à la page 57
- “Installation d'un système de fichiers racine ZFS (installation JumpStart)” à la page 63
- “Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Solaris Live Upgrade)” à la page 67
- “Prise en charge ZFS des périphériques de swap et de vidage” à la page 81
- “Initialisation à partir d'un système de fichiers racine ZFS” à la page 83

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

Vous trouverez des informations de dépannage mises à jour sur le site suivant :

http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide

Installation et initialisation d'un système de fichiers racine ZFS (présentation)

La version Solaris 10 10/08 permet d'installer et d'initialiser un système de fichiers racine ZFS des manières suivantes :

- Il est possible d'effectuer une installation initiale lorsque ZFS est sélectionné comme système de fichiers racine.

- Vous pouvez utiliser la fonction Solaris Live Upgrade pour migrer d'un système de fichiers racine UFS vers un système de fichiers racine ZFS. Vous pouvez en outre utiliser Solaris Live Upgrade pour effectuer les tâches suivantes :
 - Créer un nouvel environnement d'initialisation dans un pool racine ZFS existant.
 - Créer un nouvel environnement d'initialisation dans un nouveau pool racine ZFS.

Une fois qu'un système SPARC ou x86 est installé avec un système de fichiers racine ZFS ou migré vers un système de fichiers racine ZFS, le système s'initialise automatiquement à partir du système de fichiers racine 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 racine ZFS” à la page 83](#).

Fonctions d'installation de ZFS

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

- Le programme interactif d'installation en mode texte de Solaris vous permet d'installer un système de fichiers racine UFS ou ZFS. Le système de fichiers par défaut est toujours UFS pour cette version de Solaris. Vous pouvez accéder au programme interactif d'installation en mode texte d'une des manières suivantes :
 - Sur un système SPARC, utilisez la syntaxe suivante à partir du DVD d'installation de Solaris :

```
ok boot cdrom - text
```
 - Sur un système SPARC, utilisez la syntaxe suivante lors d'une initialisation à partir du réseau :

```
ok boot net - text
```
 - Sur un système x86, sélectionnez l'option d'installation en mode texte lorsque vous y êtes invité.
- Les fonctions JumpStart™ personnalisées vous permettent de configurer un profil pour créer un pool de stockage ZFS et désigner un système de fichiers ZFS d'initialisation.
- Vous pouvez utiliser la fonction Solaris Live Upgrade pour migrer d'un système de fichiers racine UFS vers un système de fichiers racine ZFS. Les commandes `lucreate` et `luactivate` ont été améliorées afin de prendre en charge les pools et systèmes de fichiers ZFS. Les commandes `lustatus` et `ludelete` fonctionnent comme dans les versions précédentes de Solaris.
- Vous pouvez configurer un pool racine ZFS mis en miroir en sélectionnant deux disques au cours de l'installation. Vous pouvez également créer un pool racine ZFS mis en miroir en connectant ou en ajoutant 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 racine 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 racine ZFS n'est actuellement pas disponible.
- La fonction d'installation Flash de Solaris™ permettant d'installer un système de fichiers racine ZFS n'est actuellement pas disponible.
- Le programme de mise à niveau standard ne peut pas être utilisé pour mettre à niveau votre système de fichiers racine UFS avec un système de fichiers racine ZFS. Si vous disposez d'au moins une tranche UFS d'initialisation, l'option de mise à niveau standard devrait être disponible. Si vous disposez de pools ZFS d'initialisation mais d'aucune tranche UFS d'initialisation, la seule manière d'effectuer une mise à niveau consiste à utiliser Live Upgrade à la place du programme de mise à niveau standard. Si vous disposez à la fois d'un pool ZFS d'initialisation et d'une tranche UFS d'initialisation, l'option de mise à niveau standard devrait être disponible bien que seule la tranche UFS puisse être mise à niveau.

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

Vérifiez que vous disposez de la configuration requise suivante avant d'installer un système avec un système de fichiers racine ZFS ou avant de migrer un système de fichiers racine UFS vers un système de fichiers racine ZFS :

- **Informations de version Solaris** : La capacité à installer et à initialiser à partir d'un système de fichiers racine ZFS est disponible dans Solaris 10 10/08. Pour utiliser Solaris Live Upgrade afin de migrer vers un système de fichiers racine ZFS, vous devez avoir installé Solaris 10 10/08 ou mis à niveau votre version vers Solaris 10 10/08.
- **Considérations d'un pool de stockage ZFS** : Vous pouvez créer un pool de stockage ZFS lorsque vous avez effectué une installation initiale.

Afin de vous servir de Solaris Live Upgrade pour migrer un système de fichiers racine UFS vers un système de fichiers racine ZFS, un pool de stockage ZFS doit avoir été créé avant d'effectuer l'opération `lucreate`. Afin de pouvoir être mis à niveau et initialisé, le pool de stockage ZFS doit être créé avec des tranches plutôt qu'avec des disques entiers.

En outre, le pool de stockage ZFS destiné à être le *pool racine* doit satisfaire à la configuration suivante :

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

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

- L'installation d'un système de fichiers racine ZFS requiert 768 Mo de mémoire minimum.
- 1 Go de mémoire est recommandé pour optimiser les performances globales du ZFS.
- 16 Go d'espace disque sont recommandés. L'espace est utilisé comme suit :
 - **Zone de swap et périphérique de vidage** : La zone de swap par défaut correspond à la moitié de la mémoire physique et doit être comprise entre 512 Mo et 2 Go. Le périphérique de vidage correspond à la moitié de la mémoire physique et doit être comprise entre 512 Mo et 2 Go. La taille de la zone de swap et du périphérique de vidage peut être ajustée avant, 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 82.
 - **Environnement d'initialisation** : 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 requiert environ 6 Go. Chaque environnement d'initialisation ZFS cloné à partir d'un autre environnement d'initialisation ZFS ne requiert pas d'espace disque supplémentaire ; toutefois, prenez en compte le fait que la taille de l'environnement d'initialisation est susceptible d'augmenter lors de l'application de patches. Tous les environnements d'initialisation ZFS d'un même pool racine utilisent les mêmes périphériques de swap et de vidage.

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

- Le pool doit avoir une étiquette SMI. Cette condition doit être respectée si le pool est créé avec des tranches de disque.
- Le pool doit se trouver sur une ou plusieurs tranches de disque mises en miroir mais pas dans une configuration RAID-Z ni une configuration non redondante de plusieurs disques. 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
```

- Sur un système x86, le disque doit contenir une table `fdisk`.
- Les disques désignés comme disques d'initialisation dans un pool racine ZFS doivent être limités à 1 To sur les systèmes SPARC tout comme sur les systèmes x86.

Installation d'un système de fichiers racine ZFS (installation initiale)

Cette version de Solaris vous permet d'effectuer une installation initiale à l'aide du programme interactif d'installation en mode texte de Solaris pour créer un pool de stockage ZFS contenant un système de fichiers racine ZFS d'initialisation. Si vous disposez d'un pool de stockage ZFS que vous souhaitez utiliser pour votre système de fichiers racine ZFS, servez-vous de Solaris Live Upgrade pour migrer votre système de fichiers racine UFS existant vers un système de fichiers racine ZFS dans un pool de stockage ZFS existant. Pour plus d'informations, reportez-vous à la section [“Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS \(Solaris Live Upgrade\)”](#) à la page 67.

Si vous décidez de configurer des zones après l'installation initiale d'un système de fichiers racine ZFS et vous prévoyez l'application d'un correctif au système ou sa mise à niveau, reportez-vous à [“Utilisation de Solaris Live Upgrade pour migrer un système avec zones”](#) à la page 75.

Si vous disposez déjà de pools de stockage ZFS sur votre système, ces derniers sont reconnus par le message suivant, mais restent inchangés, sauf si vous sélectionnez les disques des pools existants pour créer le nouveau 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.

Avant de lancer l'installation initiale pour créer un pool de stockage ZFS, reportez-vous à la section [“Configuration requise pour l'installation de Solaris et de Solaris Live Upgrade pour la prise en charge de ZFS”](#) à la page 55.

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation

Le processus interactif d'installation en mode texte de Solaris est le même que dans les précédentes versions de Solaris, exception faite d'un message vous invitant à créer un système de fichiers racine UFS ou ZFS. UFS demeure dans cette version le système de fichiers par défaut. Si vous sélectionnez un système de fichiers racine ZFS, un message vous invite à créer un pool de stockage ZFS. L'installation d'un système de fichiers racine ZFS implique les étapes suivantes :

1. Sélectionnez la méthode d'installation interactive de Solaris puisqu'une installation Flash de Solaris n'est pas disponible pour créer un système de fichiers racine ZFS d'initialisation.

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

Utilisez Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS si la version actuelle, Solaris 10 10/08, est déjà installée. Pour plus d'informations sur la migration vers un système de fichiers racine ZFS, reportez-vous à la section [“Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS \(Solaris Live Upgrade\)”](#) à la page 67.

2. Pour créer un système de fichiers racine ZFS, sélectionnez l'option ZFS. Exemple :

```
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 de Solaris, à l'exception du texte suivant :

```
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.
```

Vous pouvez sélectionner le ou les disques à utiliser pour le pool racine ZFS. Si vous sélectionnez deux disques, une configuration de double disque mis en miroir est définie pour le pool racine. Un pool mis en miroir double ou triple disque est optimal. Si vous disposez de huit disques et les sélectionner tous, ces huit disques sont utilisés pour le pool racine comme miroir géant. Cette configuration n'est pas optimale. Pour créer un pool racine mis en miroir composé de quatre miroirs double disque, configurez un pool double disque mis en miroir au cours de l'installation initiale, puis utilisez la commande `zpool attach` pour attacher les six autres disques une fois l'installation terminée. La configuration de pool RAID-Z n'est pas prise en charge pour le pool racine. 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 94.

4. Une fois que vous avez sélectionné un ou plusieurs disques pour le pool de stockage ZFS, un écran similaire 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
```

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

```

ZFS Root Dataset Name: s10s_u6wos_nightly
ZFS Pool Size (in MB): 34731
Size of Swap Area (in MB): 512
Size of Dump Area (in MB): 512
(Pool size must be between 6413 MB and 34731 MB)

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

```

Vous pouvez, 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 remplaçant la valeur de texte par défaut par le nouveau texte. 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 du jeu de données racine est remplacé par `zfs1008`.

```

ZFS Pool Name: rpool
ZFS Root Dataset Name: zfs1008
ZFS Pool Size (in MB): 34731
Size of Swap Area (in MB): 512
Size of Dump Area (in MB): 512
(Pool size must be between 6413 MB and 34731 MB)

```

5. Vous pouvez modifier le profil d'installation dans ce dernier écran de l'installation.
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: clt2d0
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: zfs1008

```

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

Pool Size: 34731 MB
Devices in Pool: c1t2d0

Une fois l'installation terminée, examinez les informations concernant le pool de stockage et le système de fichiers ZFS. Exemple :

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

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
c1t2d0s0	ONLINE	0	0	0

errors: No known data errors

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	5.46G	27.8G	94.5K	/rpool
rpool/ROOT	4.46G	27.8G	18K	legacy
rpool/ROOT/zfs1008	4.46G	27.8G	4.46G	/
rpool/dump	512M	27.8G	512M	-
rpool/export	38K	27.8G	20K	/export
rpool/export/home	18K	27.8G	18K	/export/home
rpool/swap	512M	28.3G	12.2M	-

L'exemple de sortie de la commande `zfs list` identifie les composants du pool racine, notamment les entrées `rpool/ROOT`, qui ne sont par défaut pas accessibles.

Si vous avez au départ créé un pool de stockage ZFS avec un disque, vous pouvez le convertir en une configuration ZFS mise en miroir une fois l'installation terminée à l'aide de la commande `zpool attach` pour y connecter un disque disponible. Exemple :

```
# zpool attach rpool c1t2d0s0 c1t3d0s0
# zpool status
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 0h0m, 5.03% done, 0h13m to go
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

```
mirror      ONLINE      0      0      0
  c1t2d0s0  ONLINE      0      0      0
  c1t3d0s0  ONLINE      0      0      0
```

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

La réargenture des données sur le nouveau disque prend un certain temps mais le pool reste disponible.

En attendant une solution au problème CR 6668666, vous devez installer les informations d'initialisation sur les disques supplémentaires que vous connectez, à l'aide de la commande `installboot` ou de la commande `installgrub` si vous souhaitez activer l'initialisation sur les autres disques du miroir. Si vous créez un pool racine ZFS mis en miroir par la méthode d'installation initiale, cette étape n'est pas nécessaire. Pour plus d'informations sur l'installation des informations d'initialisation, reportez-vous à la section [“Initialisation à partir d'un disque alternatif d'un pool racine ZFS mis en miroir”](#) à la page 83.

Pour plus d'informations sur l'ajout ou la connexion de disques, reportez-vous à la section [“Gestion de périphériques dans un pool de stockage ZFS”](#) à la page 106.

Pour créer un autre environnement d'initialisation ZFS dans le même pool de stockage, vous pouvez utiliser la commande `lucreate`. Dans l'exemple suivant, un nouvel environnement d'initialisation nommé `zfs10082BE` est créé. L'environnement d'initialisation actuel nommé `zfs1008BE` (affiché dans la sortie de la commande `zfs list`) n'est pas reconnu dans la sortie de la commande `lustatus` tant que le nouvel environnement d'initialisation n'est pas créé.

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 zfs10082BE

```
Analyzing system configuration.
Comparing source boot environment <zfs1008BE> 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 <zfs10082BE>.
Source boot environment is <zfs1008BE>.
Creating boot environment <zfs10082BE>.
Cloning file systems from boot environment <zfs1008BE> to create boot environment <zfs10082BE>.
Creating snapshot for <rpool/ROOT/zfs1008BE> on <rpool/ROOT/zfs1008BE@zfs10082BE>.
```

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

```

Creating clone for <rpool/ROOT/zfs1008BE@zfs10082BE> on <rpool/ROOT/zfs10082BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs10082BE>.
Population of boot environment <zfs10082BE> successful.
Creation of boot environment <zfs10082BE> successful.

```

La création d'un environnement d'initialisation ZFS dans le même pool fait appel aux mêmes fonctions de clonage et d'instantané ZFS afin de créer l'environnement d'initialisation instantanément. Pour plus d'informations sur l'utilisation de Solaris Live Upgrade pour une migration racine ZFS, reportez-vous à la section [“Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS \(Solaris Live Upgrade\)”](#) à la page 67.

Vérifiez ensuite les nouveaux environnements d'initialisation. Exemple :

```

# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                 Complete Now    On Reboot Delete Status
-----
ufs1008BE             yes     no     no      yes   -
zfs1008BE             yes     yes    yes     no    -
zfs10082BE           yes     no     no      yes   -
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
rpool                 5.65G 27.6G  19K   /rpool
rpool/ROOT            4.64G 27.6G  18K   /rpool/ROOT
rpool/ROOT/zfs10082BE 98.5K 27.6G  4.64G /tmp/.alt.luupdall.5312
rpool/ROOT/zfs1008BE  4.64G 27.6G  4.64G /
rpool/ROOT/zfs1008BE@zfs10082BE 92.5K - 4.64G -
rpool/dump            515M 27.6G  515M -
rpool/swap            513M 28.1G  16K -

```

Pour effectuer l'initialisation à partir d'un environnement d'initialisation alternatif, utilisez la commande `luactivate`. Après avoir activé l'environnement d'initialisation sur un système SPARC, vous pouvez utiliser la commande `boot -L` pour identifier les environnements d'initialisation disponibles lorsque le périphérique d'initialisation contient un pool de stockage ZFS. Lors de l'initialisation à partir d'un système x86, identifiez l'environnement d'initialisation à partir duquel effectuer l'initialisation dans le menu GRUB.

Par exemple, sur un système SPARC, utilisez la commande `boot -L` pour afficher une liste d'environnements d'initialisation disponibles. Pour effectuer l'initialisation à partir du nouvel environnement d'initialisation, `zfs10082BE`, sélectionnez l'option 2. Saisissez ensuite la commande `boot -Z` affichée.

```

ok boot -L
Executing last command: boot -L
Boot device: /pci@1f,0/pci@1/scsi@8/disk@1,0:a File and args: -L

```

EXEMPLE 4-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

```
1 zfs1008BE
2 zfs10082BE
Select environment to boot: [ 1 - 2 ]: 2
```

To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfs10082BE

```
Program terminated
ok boot -Z rpool/ROOT/zfs10082BE
```

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 racine ZFS”](#) à la page 83.

Installation d'un système de fichiers racine ZFS (installation JumpStart)

Vous pouvez créer un profil JumpStart pour installer un système de fichiers racine ZFS ou un système de fichiers racine UFS. Si le profil est configuré pour installer un système de fichiers racine UFS, tous les mots clés de profil existants fonctionnent comme dans les versions précédentes de Solaris.

Un profil spécifique à ZFS doit contenir le nouveau mot clé `rpool`. Le mot clé "pool" installe un nouveau pool racine et un nouvel environnement d'initialisation est par défaut créé. 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 Solaris 10 : Installation JumpStart personnalisée et installation avancée](#).

Si vous décidez de configurer des zones après l'installation JumpStart d'un système de fichiers racine ZFS et vous prévoyez l'application d'un correctif au système ou sa mise à niveau, reportez-vous à [“Utilisation de Solaris Live Upgrade pour migrer un système avec zones”](#) à la page 75.

Exemples de profils JumpStart de 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 `rpool nouveau_pool` 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` sur la base de la moitié de la mémoire physique, à concurrence de 2 Go, dans une configuration de disques mis en miroir (mot clé `mirror` et disques `c0t0d0` et `c0t1d0`). Les caractéristiques de l'environnement d'initialisation sont définies avec le mot clé `bootenv` afin d'installer un nouvel environnement d'initialisation avec le mot clé `installbe`; un environnement (bename) nommé `s10u6-xx` est créé.

```
install_type initial-install
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename s10u6-xx
```

Le profil suivant effectue une installation initiale avec le mot clé `install_type initial-install` du métacluster `SUNWCall` dans un nouveau pool de 80 Go appelé `newpool`. Ce pool est créé avec un volume de swap et un volume de vidage de 2 Go chacun, dans une configuration mise en miroir comprenant 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 avec le mot clé `bootenv` afin d'installer un nouvel environnement d'initialisation avec le mot clé `installbe`; un environnement (bename) nommé `s10u6-xx` est créé.

```
install_type initial-install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename s10u6-xx
```

Vous pouvez utiliser le profil suivant ou une syntaxe similaire pour conserver les systèmes de fichiers UFS existants sur la tranche 1 et la tranche 3, par exemple.

```
filesystems rootdisk.s1 existing ignore
filesystems rootdisk.s3 existing ignore
pool rpool auto 2G 2G rootdisk.s0
```

Vous pouvez utiliser le profil suivant ou une syntaxe similaire pour créer la tranche 1 et la tranche 3 des systèmes de fichiers UFS, par exemple.

```
filesystems rootdisk.s1 8196
filesystems rootdisk.s3 8196
pool rpool auto 2G 2G rootdisk.s0
```

Mots clés JumpStart de ZFS

Les mots clés suivants sont autorisés dans un profil spécifique à ZFS :

`auto` Spécifie la taille des tranches du pool, du volume de swap ou du volume de vidage automatiquement. La taille du disque est contrôlée pour s'assurer que la taille

minimale peut être satisfaite. Si la taille minimale peut être satisfaite, la taille de pool optimale est attribuée en fonction des contraintes, notamment de la taille des disques, des tranches conservées, etc.

Par exemple, lorsque vous spécifiez `c0t0d0s0`, une tranche de taille optimale est créée si vous spécifiez les mots clés `all` ou `auto`. Vous pouvez également indiquer une taille spécifique de tranche, de volume de swap ou de volume de vidage.

Le mot clé `auto` fonctionne de manière similaire au mot clé `all` lorsqu'il est utilisé avec un pool racine ZFS car les pools ne disposent pas du concept d'espace inutilisé.

`bootenv` Ce mot clé identifie les caractéristiques de l'environnement d'initialisation.

Le mot clé `bootenv` existe déjà mais de nouvelles options sont définies. Utilisez la syntaxe de mot clé `bootenv` suivante pour créer un environnement racine ZFS d'initialisation :

```
bootenv installbe bename nom-d'environnement-d'initialisation [ dataset
point-de-montage]
```

`installbe`

Crée un nouvel environnement d'initialisation identifié par l'option `bename` et l'entrée *nom-d'environnement-d'initialisation*, puis l'installe.

`bename nom-d'environnement-d'initialisation`

Identifie le *nom-d'environnement-d'initialisation* à installer.

Si l'option `bename` n'est pas utilisée avec le mot clé `pool`, un environnement d'initialisation par défaut est créé.

`dataset point-de-montage`

Utilisez le mot clé facultatif `dataset` pour identifier un jeu de données `/var` distinct du jeu de données racine. La valeur *point-de-montage* 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 racine à créer. La syntaxe de mot clé suivante doit être respectée :

poolname poolsize swapsize dumpsize vdevlist

poolname Identifie le nom du pool à créer. Le pool est créé d'après la *taille* de pool spécifiée et avec les périphériques physiques spécifiés(*vdevs*). L'option `poolname` ne doit pas identifier un nom de 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` signifie que la taille de pool optimale est attribuée en fonction des contraintes, notamment de la taille des disques, des tranches conservées, etc. La valeur `existing` signifie que les limites des tranches existantes de ce nom sont conservées et écrasées. L'unité de taille présumée est Mo à moins d'indiquer g (Go).

swapsize Spécifie la taille du volume de swap à créer. La valeur peut être `auto` qui signifie que la taille de swap par défaut est utilisée ou, une *taille* que vous spécifiez. L'unité de taille présumée est Mo à moins d'indiquer g (Go).

dumpsize Spécifie la taille du volume de vidage à créer. La valeur peut être `auto` qui signifie que la taille de swap par défaut est utilisée ou, une *taille* que vous spécifiez. L'unité de taille présumée est Mo à moins d'indiquer g (Go).

vdevlist Spécifie un ou plusieurs périphériques à utiliser pour créer le pool. Le format de *vdevlist* est identique à celui de la commande `zpool create`. À l'heure actuelle, seules les configurations mises en miroir sont prises en charge lorsque plusieurs périphériques sont spécifiés. Les périphériques figurant dans *vdevlist* doivent être des tranches du pool racine. La chaîne `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 94.

Problèmes liés à l'installation JumpStart d'un ZFS

Prenez en compte les problèmes suivants avant de lancer une installation JumpStart d'un système de fichiers racine ZFS d'initialisation.

- Un pool de stockage ZFS existant ne peut pas être utilisé lors d'une installation JumpStart pour créer un système de fichiers racine ZFS d'initialisation. Vous devez créer un pool de stockage ZFS conformément au type de syntaxe suivant :

```
pool rpool 20G 4G 4G c0t0d0s0
```

La ligne complète du mot clé pool est nécessaire car vous ne pouvez pas utiliser un pool existant. Exemple :

```
install_type initial_install
cluster SUNWCall
pool rpool 20G 4g 4g any
bootenv installbe bename newBE
```

- 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 de Solaris et de Solaris Live Upgrade pour la prise en charge de ZFS”](#) à la page 55. Par exemple, la syntaxe en gras n'est pas acceptable :

```
install_type initial_install
cluster SUNWCall
pool rpool all auto auto mirror c0t0d0 c0t1d0
bootenv installbe bename newBE
```

La syntaxe en gras suivante est acceptable :

```
install_type initial_install
cluster SUNWCall
pool rpool all auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename newBE
```

Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Solaris Live Upgrade)

Les fonctions précédentes de Solaris Live Upgrade sont disponibles et, si liées aux composants UFS, fonctionnent comme dans les versions précédentes de Solaris.

Voici l'ensemble des nouvelles fonctions disponibles :

- Lorsque vous migrez un système de fichiers racine UFS vers un système de fichiers racine ZFS, vous devez désigner un pool de stockage ZFS existant à l'aide de l'option -p.
- Si les composants du système de fichiers racine UFS sont répartis sur diverses tranches, ils sont migrés vers le pool racine ZFS.

- Vous pouvez migrer un système avec zones mais les configurations prises en charge sont limitées. Pour plus d'informations, reportez-vous à la section [“Utilisation de Solaris Live Upgrade pour migrer un système avec zones”](#) à la page 75.
- Solaris Live Upgrade peut faire appel aux fonctions d'instantané et de clonage ZFS lorsque vous créez un environnement d'initialisation ZFS dans le même pool. La création d'un environnement d'initialisation est de ce fait plus rapide que dans les versions précédentes de Solaris.

Pour plus d'informations sur l'installation de Solaris et les fonctions de Solaris Live Upgrade, reportez-vous au [Guide d'installation de Solaris 10 : Solaris Live Upgrade et planification de la mise à niveau](#).

Le processus de base de migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Solaris Live Upgrade) est le suivant :

- Installez Solaris 10 10/08 ou, utilisez le programme de mise à niveau standard pour mettre à niveau une version précédente de Solaris 10 sur tout système SPARC ou x86 pris en charge.
- Lorsque vous exécutez Solaris 10 10/08, créez le cas échéant un pool de stockage ZFS pour votre système de fichiers racine ZFS.
- Utilisez Solaris Live Upgrade pour migrer votre système de fichiers racine UFS vers un système de fichiers racine ZFS.
- Activez votre environnement d'initialisation ZFS à l'aide de la commande `luactivate`.

Pour plus d'informations sur les exigences de ZFS et de Solaris Live Upgrade, reportez-vous à la section [“Configuration requise pour l'installation de Solaris et de Solaris Live Upgrade pour la prise en charge de ZFS”](#) à la page 55.

Problèmes de migration d'un ZFS avec Solaris Live Upgrade

Consultez la liste de problèmes suivante avant d'utiliser Solaris Live Upgrade pour migrer votre système de fichiers racine UFS vers un système de fichiers racine ZFS :

- L'option de mise à niveau standard de l'IG d'installation de Solaris n'est pas disponible pour la migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS. Pour migrer un système de fichiers UFS, vous devez utiliser Solaris Live Upgrade.
- Vous devez créer le pool de stockage ZFS à utiliser pour l'initialisation avant de lancer Solaris Live Upgrade. En outre, en raison des limitations actuelles de l'initialisation, le pool racine ZFS doit être créé avec des tranches plutôt qu'avec des disques entiers. 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, vérifiez que le processus d'étiquetage n'a pas modifié le schéma du partitionnement. Dans la plupart des cas, la majeure partie de la capacité du disque doit se trouver dans les tranches destinées au pool racine.

- Vous ne pouvez pas utiliser 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 la fonction Solaris Live Upgrade ne tient pas compte du changement 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.
- Solaris Live Upgrade crée les jeux de données pour l'environnement d'initialisation, ainsi que les volumes ZFS de la zone de swap et du périphérique de vidage mais ne prend pas en compte les modifications de propriétés de jeux de données existants. Pour cette raison, pour activer une propriété de jeu de données dans le nouvel environnement d'initialisation, vous devez définir cette propriété avant l'exécution de la commande `lucreate`. Exemple :

```
# zfs set compression=on rpool/ROOT
```

- Lorsque vous créez un environnement d'initialisation alternatif 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 Solaris Live Upgrade pour mettre à niveau votre système de fichiers racine UFS vers un système de fichiers racine ZFS, vous ne pouvez pas vous servir de Solaris Live Upgrade pour mettre à niveau des systèmes de fichiers partagés ni des systèmes de fichiers qui ne sont pas des systèmes de fichiers racine.
- Vous ne pouvez pas utiliser une commande `lu` pour créer ou migrer un système de fichiers racine ZFS.

Utilisation de Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS (sans zones)

Les exemples suivants illustrent la procédure de migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS. Si vous migrez un système avec zones, reportez-vous à [“Utilisation de Solaris Live Upgrade pour migrer un système avec zones”](#) à la page 75.

EXEMPLE 4-2 Utilisation de Solaris Live Upgrade pour migrer un système de fichiers racine UFS vers un système de fichiers racine ZFS

L'exemple suivant illustre la procédure de création d'un environnement d'initialisation d'un système de fichiers racine ZFS à partir d'un système de fichiers racine UFS. L'environnement d'initialisation actuel, `ufs1008BE`, qui contient un système de fichiers racine 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, `zfs1008BE`, est identifié par l'option `-n`. L'utilisation de la commande `lucreate` requiert l'existence préalable d'un pool de stockage ZFS.

Afin de pouvoir être mis à niveau et initialisé, le pool de stockage ZFS doit être créé avec des tranches plutôt qu'avec 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, vérifiez que le processus d'étiquetage n'a pas modifié le schéma du partitionnement. Dans la plupart des cas, la majeure partie de la capacité du disque doit se trouver dans les tranches destinées au pool racine.

```
# zpool create mpool mirror c1t0d0s0 c1t1d0s0
# lucreate -c ufs1008BE -n zfs1008BE -p mpool
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <ufs1008BE>.
Creating initial configuration for primary boot environment <zfs1008BE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufs1008BE> PBE Boot Device </dev/dsk/c0t1d0s0>.
```

EXEMPLE 4-2 Utilisation de Solaris Live Upgrade pour migrer un système de fichiers racine UFS vers un système de fichiers racine ZFS (Suite)

```

Comparing source boot environment <ufs1008BE> 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/c0t0d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfs1008BE>.
Source boot environment is <ufs1008BE>.
Creating boot environment <zfs1008BE>.
Creating file systems on boot environment <zfs1008BE>.
Creating <zfs> file system for </> in zone <global> on <mpool/ROOT/zfs1008BE>.
Populating file systems on boot environment <zfs1008BE>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Creating compare databases for boot environment <zfs1008BE>.
Creating compare database for file system </mpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfs1008BE>.
Making boot environment <zfs1008BE> bootable.
Creating boot_archive for /.alt.tmp.b-zv.mnt
updating /.alt.tmp.b-zv.mnt/platform/sun4u/boot_archive
Population of boot environment <zfs1008BE> successful.
Creation of boot environment <zfs1008BE> successful.

```

Une fois l'exécution de la commande `lucreate` terminée, utilisez la commande `lustatus` pour afficher l'état de l'environnement d'initialisation. Exemple :

```

# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
ufs1008BE             yes     yes   yes     no    -
zfs1008BE             yes     no    no      yes   -

```

Examinez ensuite la liste des composants ZFS. Exemple :

```

# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
mpool                5.64G 27.6G   19K    /mpool
mpool/ROOT           4.64G 27.6G   18K    /mpool/ROOT
mpool/ROOT/zfs1008BE 4.64G 27.6G  4.64G  /tmp/.alt.luupdall.1551

```

EXEMPLE 4-2 Utilisation de Solaris Live Upgrade pour migrer un système de fichiers racine UFS vers un système de fichiers racine ZFS (Suite)

```
mpool/dump          513M 28.1G 16K -
mpool/swap          513M 28.1G 16K -
```

Utilisez ensuite la commande `luactivate` pour activer le nouvel environnement d'initialisation ZFS. Par exemple :

```
# luactivate zfs1008BE
```

```
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs1008BE>.
```

```
*****
```

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.

```
*****
```

In case of a failure while booting to the target BE, the following process needs to be followed to fallback to the currently working boot environment:

1. Enter the PROM monitor (ok prompt).
2. Change the boot device back to the original boot environment by typing:

```
setenv boot-device /pci@1f,0/pci@1/scsi@8/disk@0,0:a
```

3. Boot to the original boot environment by typing:

```
boot
```

```
*****
```

```
Modifying boot archive service
```

```
Activation of boot environment <zfs1008BE> successful.
```

Réinitialisez ensuite le système afin d'utiliser l'environnement d'initialisation ZFS.

```
# init 6
```

```
# svc.startd: The system is coming down. Please wait.
```

```
svc.startd: 79 system services are now being stopped.
```

```
.
.
.
```

EXEMPLE 4-2 Utilisation de Solaris Live Upgrade pour migrer un système de fichiers racine UFS vers un système de fichiers racine ZFS (Suite)

Confirmez que l'environnement d'initialisation ZFS est actif.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
ufs1008BE             yes     no     no      yes   -
zfs1008BE             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. Un messages du type suivant s'affiche lorsque vous repassez à l'environnement d'initialisation UFS.

```
# luactivate ufs1008BE
WARNING: The following files have changed on both the current boot
environment <zfs1008BE> zone <global> and the boot environment to be
activated <ufs1008BE>:
    /etc/zfs/zpool.cache
INFORMATION: The files listed above are in conflict between the current
boot environment <zfs1008BE> zone <global> and the boot environment to be
activated <ufs1008BE>. These files will not be automatically synchronized
from the current boot environment <zfs1008BE> when boot environment
<ufs1008BE> is activated.
```

Lorsque l'environnement d'initialisation UFS est obsolète, vous pouvez le supprimer à l'aide de la commande `ludelete`.

EXEMPLE 4-3 Utilisation de Solaris 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 réside sur le même pool ZFS, `mpool`, par exemple, l'option `-p` est omise.

Si vous disposez de plusieurs environnements d'initialisation ZFS sur un système SPARC, vous pouvez utiliser la commande `boot -L` pour identifier les environnements d'initialisation disponibles et sélectionner un environnement d'initialisation à partir duquel effectuer l'initialisation à l'aide de la commande `boot -Z`. Sur un système x86, vous pouvez sélectionner un environnement d'initialisation à partir du menu GRUB. Pour plus d'informations, reportez-vous à l'[Exemple 4-6](#).

EXEMPLE 4-3 Utilisation de Solaris Live Upgrade pour créer un environnement d'initialisation ZFS à partir d'un environnement d'initialisation ZFS (Suite)

```
# lucreate -n zfs10082BE
Analyzing system configuration.
Comparing source boot environment <zfs1008BE> 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 <zfs10082BE>.
Source boot environment is <zfs1008BE>.
Creating boot environment <zfs10082BE>.
Cloning file systems from boot environment <zfs1008BE> to create boot environment <zfs10082BE>.
Creating snapshot for <mpool/ROOT/zfs1008BE> on <mpool/ROOT/zfs1008BE6@zfs10082BE>.
Creating clone for <mpool/ROOT/zfs1008BE@zfs10082BE> on <mpool/ROOT/zfs10082BE>.
Setting canmount=noauto for </> in zone <global> on <mpool/ROOT/zfs10082BE>.
Population of boot environment <zfs10082BE> successful.
Creation of boot environment <zfs10082BE> successful.
```

EXEMPLE 4-4 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 patches 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 à l'aide de la commande `luupgrade` pour ajouter des packages ou des patches.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
zfs1008BE             yes     no     no       yes   -
zfs10082BE            yes     yes    yes      no    -
# luupgrade -p -n zfs1008BE -s /net/system/export/s1008/Solaris_10/Product SUNWchxge

Validating the contents of the media </net/system/export/s1008//Solaris_10/Product>.
Mounting the BE <zfs1008BE>.
Adding packages to the BE <zfs1008BE>.

Processing package instance <SUNWchxge> from </net/system/export/s1008/Solaris_10/Product>
```

EXEMPLE 4-4 Mise à niveau de votre environnement d'initialisation ZFS (luupgrade) *(Suite)*

```
Chelsio N110 10GE NIC Driver(sparc) 11.10.0,REV=2006.02.15.20.41
Copyright 2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Using </a> as the package base directory.
## Processing package information.
## Processing system information.
   3 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.
394 blocks
## Executing postinstall script.
Reboot client to install driver.
```

```
Installation of <SUNWchxge> was successful.
Unmounting the BE <zfs1008BE>.
The package add to the BE <zfs1008BE> completed.
```

Utilisation de Solaris Live Upgrade pour migrer un système avec zones

Vous pouvez utiliser Solaris Live Upgrade pour migrer un système avec zones. Les configurations prises en charge sont toutefois limitées.

Cette section décrit comment configurer et installer un système avec zones de manière à ce qu'il soit mis à jour et corrigé avec Solaris Live Upgrade. Si vous migrez vers un système de fichiers racine ZFS sans zones, reportez-vous à [“Utilisation de Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS \(sans zones\)”](#) à la page 70.

Si vous migrez un système avec zones ou envisagez de configurer un système avec zones, consultez les procédures suivantes :

- “Comment migrer un système de fichiers racine UFS avec racines de zone sur UFS vers un système de fichiers racine ZFS” à la page 76
- “Comment configurer un système de fichiers racine ZFS avec racines de zone sur ZFS” à la page 78
- “Comment mettre à niveau ou corriger un système de fichiers racine ZFS avec racines de zone sur ZFS” à la page 79
- “Résolution des problèmes de point de montage ZFS responsables de l'échec de l'initialisation” à la page 88

Suivez les procédures recommandées pour configurer des zones sur un système avec système de fichiers racine ZFS pour vérifier que vous pouvez utiliser Live Upgrade sur ce système.

▼ **Comment migrer un système de fichiers racine UFS avec racines de zone sur UFS vers un système de fichiers racine ZFS**

Suivez les étapes ci-dessous pour migrer un système de fichiers racine UFS avec zones installées vers un système de fichiers racine ZFS et une configuration de racine de zone ZFS pouvant être mis à niveau ou corrigés.

Dans les étapes suivantes, le nom du pool est `rpool` et le nom de l'environnement d'initialisation actuellement actif est `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 de Solaris 10 : Solaris Live Upgrade et planification de la mise à niveau](#).

2 Créez un pool racine.

Pour plus d'informations sur les exigences du pool racine, reportez-vous à la “[Configuration requise pour l'installation de Solaris et de Solaris Live Upgrade pour la prise en charge de ZFS](#)” à la page 55.

3 Confirmez que les zones de l'environnement UFS sont initialisées.

4 Créez le nouvel environnement d'initialisation.

```
# lucreate -n S10BE2 -p rpool
```

Cette commande crée des jeux de données dans le pool racine 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.

```
# luactivate s10BE2
```

Le système exécute maintenant un système de fichiers racine ZFS mais les racines de zone sur UFS se trouvent toujours sur le système de fichiers racine UFS. Les étapes suivantes sont nécessaires pour finaliser la migration des zones UFS vers une configuration ZFS prise en charge.

6 Redémarrez 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 dans le pool.

```
# lucreate S10BE3
```

c. Activez le nouvel environnement d'initialisation.

```
# luactivate S10BE3
```

d. Redémarrez le système.

```
# init 6
```

Cette étape vérifie que l'environnement d'initialisation ZFS et les zones ont été initialisés.

8 Résolez tout problème potentiel de point de montage de cette version Solaris.

Il est possible qu'un bogue dans la fonctionnalité Live Upgrade provoque l'échec de l'initialisation de l'environnement d'initialisation non actif. 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 de zone de l'environnement d'initialisation.

a. Contrôlez la sortie `zfs list`.

Vérifiez qu'elle ne contient aucun point de montage temporaire erroné. Exemple :

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10u6
```

NAME	MOUNTPOINT
rpool/ROOT/s10u6	/.alt.tmp.b-VP.mnt/
rpool/ROOT/s10u6/zones	/.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10u6/zones/zonerootA	/.alt.tmp.b-VP.mnt/zones/zonerootA

Le point de montage pour l'environnement d'initialisation ZFS racine (rpool/ROOT/s10u6) doit être /.

- b. Réinitialisez les points de montage pour l'environnement d'initialisation ZFS et ses jeux de données.

Exemple :

```
# zfs inherit -r mountpoint rpool/ROOT/s10u6
# zfs set mountpoint=/ rpool/ROOT/s10u6
```

- c. Redémarrez le système.

Lorsque vous avez la possibilité d'initialiser un environnement d'initialisation spécifique, soit par le biais du menu GRUB, soit à l'invite OpenBoot Prom, sélectionnez l'environnement d'initialisation dont les points de montage viennent d'être corrigés.

▼ Comment configurer un système de fichiers racine ZFS avec racines de zone sur ZFS

Suivez les étapes ci-dessous pour installer un système de fichiers racine ZFS et une configuration de racine de zone ZFS pouvant être mis à niveau ou corrigés. Dans cette configuration, les racines de zone ZFS sont créées sous forme de jeux de données ZFS.

Dans les étapes suivantes, le nom du pool est `rpool` et le nom de l'environnement d'initialisation actuellement actif est `S10be`.

- 1 **Installez le système avec une racine ZFS en utilisant soit la méthode interactive d'installation initiale, soit la méthode d'installation Solaris JumpStart.**

Pour plus d'informations sur l'installation d'un système de fichiers racine ZFS en utilisant la méthode d'installation initiale ou la méthode Solaris JumpStart, reportez-vous à [“Installation d'un système de fichiers racine ZFS \(installation initiale\)”](#) à la page 57 ou à [“Installation d'un système de fichiers racine ZFS \(installation JumpStart\)”](#) à la page 63.

- 2 **Initialisez le système à partir du pool racine nouvellement créé.**

- 3 **Créez un jeu de données pour le regroupement des racines de zone.**

Exemple :

```
# zfs create -o canmount=noauto rpool/ROOT/S10be/zones
```

Le nom du jeu de données de zones peut être tout nom de jeu de données légal. Dans les étapes suivantes, le nom du jeu de données est `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 Solaris Live Upgrade et le code de démarrage du système.

- 4 **Montez le jeu de données de conteneur de zones nouvellement 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 racine de zone.

```
# zfs create -o canmount=noauto rpool/ROOT/S10be/zones/zonerootA
# zfs mount rpool/ROOT/S10be/zones/zonerootA
```

6 Définissez les droits appropriés dans le répertoire de racine de 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 utilisant la syntaxe suivante :

```
zonecfg:zoneA> set autoboot=true
```

8 Installez la zone.

```
# zoneadm -z zoneA install
```

9 Initialisez la zone.

```
# zoneadm -z zoneA boot
```

▼ Comment mettre à niveau ou corriger un système de fichiers racine ZFS avec racines de zone sur ZFS

Procédez aux étapes suivantes pour mettre à niveau ou corriger le système de fichiers racine ZFS avec racines de zone sur ZFS. Ces mises à jour peuvent être une mise à niveau du système ou l'application de correctifs.

Dans les étapes suivantes, newBE est le nom de l'environnement d'initialisation mis à niveau ou corrigé.

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. De nouveaux jeux de données sont créés pour chaque jeu de données de l'environnement d'initialisation d'origine. Ils sont créés dans le même pool que le pool racine actuel.

2 Sélectionnez l'une des options suivantes pour mettre à niveau le système ou appliquer les correctifs au nouvel environnement d'initialisation.

- Mettez à niveau le système.

```
# luupgrade -u -n newBE -s /net/install/export/s10u7/latest
```

L'option -s représente l'emplacement d'un mode d'installation Solaris.

- Appliquez les correctifs au nouvel environnement d'initialisation.

```
# luupgrade -t -n newBE -t -s /patchdir 139147-02 157347-14
```

3 Activez le nouvel environnement d'initialisation une fois que ses mises à jour ont été effectuées.

```
# luactivate newBE
```

4 Initialisez à partir de l'environnement d'initialisation nouvellement activé.

```
# init 6
```

5 Résolvez tout problème potentiel de point de montage de cette version Solaris.

Il est possible qu'un bogue dans la fonctionnalité Live Upgrade provoque l'échec de l'initialisation de l'environnement d'initialisation non actif. 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 de zone de l'environnement d'initialisation.

a. Contrôlez la sortie `zfs list`.

Vérifiez qu'elle ne contient aucun point de montage temporaire erroné. 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 racine ZFS (rpool/ROOT/newBE) doit être /.

b. Réinitialisez les points de montage pour l'environnement d'initialisation ZFS et ses jeux de données.

Exemple :

```
# zfs inherit -r mountpoint rpool/ROOT/newBE
# zfs set mountpoint=/ rpool/ROOT/newBE
```

c. Redémarrez le système.

Lorsque vous avez la possibilité d'initialiser un environnement d'initialisation spécifique, soit par le biais du menu GRUB, soit à l'invite OpenBoot Prom, sélectionnez l'environnement d'initialisation dont les points de montage viennent d'être corrigés.

Prise en charge ZFS des périphériques de swap et de vidage

Lors d'une installation initiale ou de l'utilisation de Solaris Live Upgrade à partir d'un système de fichiers UFS, une zone de swap est créée sur un volume ZFS du pool racine ZFS. La taille de la zone de swap correspond à la moitié de la mémoire physique et doit être comprise entre 512 Mo et 2 Go. Par exemple :

```
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/mpool/swap 253,3      16  8257520  8257520
```

Lors d'une installation initiale ou de l'utilisation de Solaris Live Upgrade à partir d'un système de fichiers UFS, un périphérique de vidage est créé sur un volume ZFS du pool racine ZFS. La taille du périphérique de vidage correspond à la moitié de la mémoire physique et doit être comprise entre 512 Mo et 2 Go. Le périphérique de vidage ne nécessite aucune administration une fois configuré. Exemple :

```
# dumpadm
  Dump content: kernel pages
  Dump device: /dev/zvol/dsk/mpool/dump (dedicated)
Savecore directory: /var/crash/t2000
  Savecore enabled: yes
```

Prenez en compte les points suivants lorsque vous travaillez avec des périphériques de swap et de vidage ZFS :

- Vous devez utiliser des volumes ZFS distincts pour les périphériques de swap et de vidage.
- L'utilisation d'un fichier swap sur un système de fichiers ZFS n'est actuellement pas prise en charge.
- En raison de CR 6724860, vous devez exécuter la commande `savecore` manuellement pour enregistrer un vidage mémoire sur incident lorsque vous utilisez un volume de vidage ZFS.
- 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 Solaris précédentes. Pour plus d'informations, reportez-vous au [Chapitre 20, “Configuring Additional Swap Space \(Tasks\)”](#) du *System Administration Guide: Devices and File Systems* et au [Chapitre 17, “Managing System Crash Information \(Tasks\)”](#) du *System Administration Guide: Advanced Administration* (tous deux en anglais).

Ajustement de la taille de vos périphériques de swap et de vidage ZFS

La façon dont une installation racine ZFS attribue une taille diffère selon qu'il s'agit d'un périphérique de swap ou d'un périphérique de vidage ; il s'avère pour cela parfois nécessaire d'ajuster la taille des périphériques de swap et de vidage avant, pendant ou après l'installation.

- 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 d'utiliser Solaris Live Upgrade. Exemple :

1. Créer le pool de stockage.

```
# zpool create rpool mirror c0t0d0s0 c0t1d0s0
```

2. Définissez la taille du périphérique de vidage. Configurez la taille du bloc sur 128 Ko.

```
# zfs create -V 2G -b 128k rpool/dump
```

3. Sélectionnez l'une des options suivantes pour créer la zone de swap :

- Sur un système SPARC, définissez une taille pour la zone de swap. Configurez la taille du bloc sur 8 Ko.

```
# zfs create -V 2G -b 8k rpool/swap
```

- Sur un système x86, définissez une taille pour la zone de swap. Configurez la taille du bloc sur 4 Ko.

```
# zfs create -V 2G -b 4k rpool/swap
```

Solaris Live Upgrade ne redimensionne pas des volumes de swap et de vidage existants.

- Vous pouvez rétablir la propriété `volsize` des périphériques de swap et de vidage après l'installation d'un système. Exemple :

```
# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME          PROPERTY  VALUE      SOURCE
rpool/dump    volsize   2G         -
```

- 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 entrées 2g et 2g attribuent 2 Go comme taille de zone de swap et comme taille de périphérique de vidage.

Initialisation à partir d'un système de fichiers racine 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 racine ZFS, les noms de chemin du fichier archive et du fichier noyau sont résolus dans le système de fichiers racine sélectionné pour l'initialisation.

Lorsque le système est initialisé pour l'installation, un disque RAM est utilisé pour le système de fichiers racine pendant toute la procédure d'installation afin de ne pas avoir à effectuer l'initialisation à partir d'un média amovible.

Si vous procédez à une installation initiale de Solaris 10 10/08 ou utilisez Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS dans cette version, vous pouvez effectuer l'initialisation à partir d'un système de fichiers racine ZFS sur un système SPARC tout comme sur un système x86.

L'initialisation à partir d'un système de fichiers ZFS diffère de celle d'un système de fichiers UFS car avec ZFS, un spécificateur de périphérique identifie un pool de stockage par opposition à un seul système de fichiers racine. Un pool de stockage peut contenir plusieurs *jeux de données d'initialisation* ou systèmes de fichiers racine ZFS. Lorsque vous initialisez un système à partir de ZFS, vous devez spécifier un périphérique d'initialisation et un système de fichiers racine contenu dans le pool qui a été identifié par le périphérique d'initialisation.

Par défaut, le jeu de données sélectionné pour l'initialisation est celui qui est identifié par la propriété `boot fs` du pool. Cette sélection par défaut peut être remplacée en spécifiant un jeu de données d'initialisation alternatif qui est inclus dans la commande `boot -Z`.

Initialisation à partir d'un disque alternatif d'un pool racine ZFS mis en miroir

Vous pouvez créer un pool racine ZFS mis en miroir lors de l'installation du système ou pouvez connecter un disque pour créer un pool racine ZFS mis en miroir après l'installation. Consultez les problèmes connus suivants relatifs aux pools racine ZFS mis en miroir :

- CR 6704717 – Ne mettez pas le disque principal d'une configuration racine ZFS mise en miroir hors ligne. Si la mise hors ligne ou la déconnexion d'un disque racine mis en miroir est nécessaire pour le remplacer, effectuez l'initialisation à partir d'un autre disque mis en miroir du pool.

- CR 6668666 – Vous devez installer les informations d'initialisation sur les disques supplémentaires que vous connectez, à l'aide de la commande `installboot` ou de la commande `installgrub` si vous souhaitez activer l'initialisation sur les autres disques du miroir. Si vous créez un pool racine ZFS mis en miroir par la méthode d'installation initiale, cette étape n'est pas nécessaire. Par exemple, si `c0t1d0s0` est le deuxième disque ajouté au miroir, la commande `installboot` ou la commande `installgrub` est exécutée comme suit :

```
sparc# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c0t1d0s0
```

```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c0t1d0s0
```

Vous pouvez effectuer l'initialisation à partir de divers périphériques d'un pool racine 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`) de ce pool.

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

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c1t0d0s0	ONLINE	0	0	0
c1t1d0s0	ONLINE	0	0	0

Sur un système SPARC, saisissez le disque alternatif à l'invite `ok`.

```
ok boot /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@1
```

Une fois le système réinitialisé, confirmez le périphérique d'initialisation actif. Exemple :

```
# prtconf -vp | grep bootpath
bootpath: '/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@1,0:a'
```

Sur un système x86, sélectionnez un disque alternatif dans le pool racine ZFS mis en miroir dans le menu approprié du BIOS.

Initialisation à partir d'un système de fichiers racine ZFS sur un système SPARC

Sur un système SPARC avec environnements d'initialisation multiples ZFS, vous pouvez initialiser à partir de tout environnement d'initialisation en utilisant la commande `luactivate`. Après avoir activé l'environnement d'initialisation, vous pouvez utiliser la commande `boot -L` pour afficher la liste des environnements d'initialisation lorsque le périphérique d'initialisation contient un pool de stockage ZFS.

Au cours de l'installation et de la procédure de Solaris Live Upgrade, le système de fichiers racine ZFS est automatiquement désigné avec la propriété `bootfs`.

Un pool peut contenir plusieurs jeux de données d'initialisation. Par défaut, l'entrée du jeu de données d'initialisation figurant dans le fichier `/nom-du-pool/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 racine du pool.

Lorsqu'un système est installé avec un système de fichiers racine ZFS ou est migré vers un système de fichiers racine ZFS, une entrée du type suivant est ajoutée au fichier `menu.lst` :

```
title zfs1008BE
bootfs mpool/ROOT/zfs1008BE
```

Lorsqu'un nouvel environnement d'initialisation est créé, le fichier `menu.lst` est mis à jour. En attendant une solution au problème CR 6696226, vous devez mettre à jour le fichier `menu.lst` manuellement une fois que vous avez activé l'environnement d'initialisation à l'aide de la commande `luactivate`.

Sur un système SPARC, deux nouvelles options d'initialisation sont disponibles :

- Vous pouvez utiliser la commande d'initialisation `-L` pour afficher une liste de jeux de données d'initialisation contenus dans un pool ZFS. Vous pouvez ensuite sélectionner un des jeux de données d'initialisation 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. Cette option n'est disponible que lorsque le périphérique d'initialisation contient un pool de stockage ZFS.
- Utilisez la commande `-Z dataset` pour initialiser un jeu de données ZFS spécifique.

EXEMPLE 4-5 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.

EXEMPLE 4-5 Initialisation à partir d'un environnement d'initialisation ZFS spécifique (Suite)

Par exemple, les environnements d'initialisation ZFS suivants sont disponibles comme décrit par la sortie de `lustatus` :

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
zfs1008BE             yes     yes   yes     no    -
zfs10082BE            yes     no    no      yes   -
```

Si vous disposez de plusieurs environnements d'initialisation ZFS sur votre système SPARC, vous pouvez utiliser la commande `boot -L`. Exemple :

```
ok boot -L
Executing last command: boot -L
Boot device: /pci@1f,0/pci@1/scsi@8/disk@1,0:a File and args: -L
1 zfs1008BE
2 zfs10082BE
Select environment to boot: [ 1 - 2 ]: 2

To boot the selected entry, invoke:
boot [<root-device>] -Z mpool/ROOT/zfs10082BE

Program terminated
ok boot -Z mpool/ROOT/zfs10082BE
```

EXEMPLE 4-6 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 . Exemple :

```
ok boot -F failsafe
```

Pour initialiser une archive de secours à partir d'un jeu de données d'initialisation ZFS donné, employez une syntaxe du type suivant :

```
ok boot -Z mpool/ROOT/zfs1008BE -F failsafe
```

Initialisation à partir d'un système de fichiers racine ZFS sur un système x86

Les entrées suivantes sont ajoutées au fichier `/nom-de-pool/boot/grub/menu.lst` au cours du processus d'installation ou du lancement de Solaris Live Upgrade pour initialiser ZFS automatiquement :

```
findroot (pool_mpool,0,a)
bootfs mpool/ROOT/zfs1008BE
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
```

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 racine correspondant à cette entrée de menu est un jeu de données ZFS, l'option suivante est ajoutée.

```
-B $ZFS-BOOTFS
```

EXEMPLE 4-7 x86 : Initialisation d'un système de fichiers ZFS

Lorsque vous effectuez l'initialisation à partir d'un système de fichiers ZFS, le périphérique racine est spécifié par le paramètre d'initialisation `-B $ZFS-BOOTFS` sur la ligne `kernel` ou `module` de l'entrée du menu GRUB. Cette valeur, tout comme tous les paramètres spécifiés par l'option `-B`, est transmise par GRUB au noyau. Exemple :

```
findroot (pool_mpool,0,a)
bootfs mpool/ROOT/zfs1008BE
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
```

EXEMPLE 4-8 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. Exemple :

```
title Solaris failsafe
bootfs mpool/ROOT/zfs1008BE
findroot (pool_mpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe
```

Résolution des problèmes de point de montage ZFS responsables de l'échec de l'initialisation

Le meilleur moyen de changer d'environnement d'initialisation actif est d'utiliser la commande `luactivate`. En cas d'échec de l'initialisation de l'environnement actif, qu'il soit dû à un correctif défectueux ou à une erreur de configuration, le seul moyen d'initialiser un environnement différent consiste à le sélectionner lors de l'initialisation. Vous pouvez sélectionner un environnement d'initialisation différent dans le menu GRUB sur un système x86 ou l'initialiser à partir de PROM sur un système SPARC.

Il est possible qu'un bogue dans la fonctionnalité Live Upgrade provoque l'échec de l'initialisation de l'environnement d'initialisation non actif. 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 de 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.

Les points de montage peuvent être corrigés en procédant aux étapes suivantes :

▼ Comment résoudre les problèmes de point de montage ZFS

1 Initialisez le système à partir d'une archive de secours.

2 Importez le pool.

Exemple :

```
# zpool import rpool
```

3 Une fois le pool importé, contrôlez la sortie `zfs list`.

Vérifiez qu'elle ne contient aucun point de montage temporaire erroné. Exemple :

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10u6
```

NAME	MOUNTPOINT
rpool/ROOT/s10u6	/.alt.tmp.b-VP.mnt/
rpool/ROOT/s10u6/zones	/.alt.tmp.b-VP.mnt/zones
rpool/ROOT/s10u6/zones/zonerootA	/.alt.tmp.b-VP.mnt/zones/zonerootA

Le point de montage pour l'environnement d'initialisation racine (`rpool/ROOT/s10u6`) 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.

Exemple :

```
# zfs inherit -r mountpoint rpool/ROOT/s10u6  
# zfs set mountpoint=/ rpool/ROOT/s10u6
```

5 Redémarrez le système.

Lorsque vous avez la possibilité d'initialiser un environnement d'initialisation spécifique, soit par le biais du menu GRUB, soit à l'invite OpenBoot Prom, sélectionnez l'environnement d'initialisation dont les points de montage viennent d'être corrigés.

Gestion des pools de stockage ZFS

Ce chapitre décrit la procédure de création et d'administration des pools de stockage ZFS.

Il contient les sections suivantes :

- “Composants d'un pool de stockage ZFS” à la page 91
- “Création et destruction de pools de stockage ZFS” à la page 96
- “Gestion de périphériques dans un pool de stockage ZFS” à la page 106
- “Gestion des propriétés de pool de stockage ZFS” à la page 120
- “Requête d'état de pool de stockage ZFS” à la page 123
- “Migration de pools de stockage ZFS” à la page 130
- “Mise à niveau de pools de stockage ZFS” à la page 137

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 91
- “Utilisation de tranches dans un pool de stockage ZFS” à la page 93
- “Utilisation de fichiers dans un pool de stockage ZFS” à la page 93

Utilisation de disques dans un pool de stockage ZFS

Le composant de base d'un pool de stockage est un élément de stockage physique. Le stockage physique peut être constitué de tout périphérique en mode bloc d'une taille supérieure à 128 Mo. En général, ce périphérique est un disque dur que le système peut voir dans le répertoire `/dev/dsk` .

Un disque entier (`c1t0d0`) ou une tranche individuelle (`c0t0d0s7`) peuvent constituer un périphérique de stockage. Le mode opérationnel recommandé 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: 71670953 + 16384 (reserved sectors)

Part	Tag	Flag	First Sector	Size	Last Sector
0	usr	wm	34	34.18GB	71670953
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
7	unassigned	wm	0	0	0
8	reserved	wm	71670954	8.00MB	71687337

Pour utiliser des disques entiers, ces disques doivent être nommés selon la convention Solaris standard, par exemple `/dev/dsk/cXtXdXsX`. 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.

ZFS applique une étiquette EFI lorsque vous créez un pool de stockage avec des disques entiers.

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`
- `c0t0d6s2`
- `/dev/foo/disk`

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 des configurations 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. Bien que ZFS fonctionne correctement sur de tels périphériques, les performances risquent d'être médiocres.

Pour obtenir des informations supplémentaires sur les recommandations de pools de stockage, consultez le site des pratiques ZFS recommandées :

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

Les disques sont identifiés par leur chemin et par l'ID de leur périphérique, s'il est disponible. Cette méthode permet de reconfigurer les périphériques dans un système sans requérir la mise à jour de tout état ZFS. Si un disque est déplacé du contrôleur 1 au contrôleur 2, ZFS utilise l'ID du périphérique pour déterminer si le disque a été déplacé et si son accès s'effectue à partir du contrôleur 2. L'ID du périphérique est unique pour le microprogramme du disque. Certaines mises à jour de microprogrammes sont susceptibles de modifier l'ID du périphérique, mais il est peu probable que cela se produise. Si cela se produit, ZFS peut tout de même accéder au périphérique à l'aide de son chemin et mettre à jour automatiquement l'ID de périphérique stocké. Si, par inadvertance, vous modifiez le chemin et l'ID du périphérique, exportez et réimportez le pool pour pouvoir l'utiliser.

Utilisation de tranches dans un pool de stockage ZFS

Les disques peuvent être étiquetés avec une étiquette VTOC Solaris classique lorsque vous créez un pool de stockage avec une tranche de disque.

Pour un pool racine ZFS d'initialisation, les disques du pool doivent contenir des tranches. La plus simple configuration consiste à placer toute la capacité du disque dans la tranche 0 et à utiliser cette tranche pour le pool racine.

Si vous envisagez d'utiliser des tranches pour un pool de stockage ZFS qui n'est pas un périphérique d'initialisation, examinez les conditions suivantes lorsque l'utilisation de tranches risque d'être nécessaire :

- Le nom du périphérique n'est pas standard.
- ZFS et un autre système de fichier (UFS, par exemple) partagent un disque unique.
- Un disque est utilisé en tant que swap ou périphérique de vidage.

Utilisation de fichiers dans un pool de stockage ZFS

ZFS permet également d'utiliser des fichiers UFS 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. En effet, **toute utilisation de fichier repose sur le système de fichier sous-jacent pour la cohérence**. Si vous créez un pool ZFS à partir de fichiers stockés sur un système de fichiers UFS, la garantie d'une sémantique synchrone et juste repose entièrement sur UFS.

Cependant, les fichiers peuvent s'avérer utiles lorsque vous employez ZFS pour la première fois ou en cas de disposition complexe, parce que 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. Si un fichier est déplacé ou renommé, le pool doit être exporté et réimporté pour pouvoir être utilisé, car aucun ID de périphérique n'est associé aux fichiers pour permettre de les localiser.

Fonctions de réplication d'un pool de stockage ZFS

ZFS offre une redondance des données, ainsi que des propriétés d'autorétablissement, dans une configuration mise en miroir et une configuration RAID-Z.

- [“Configuration de pool de stockage mis en miroir” à la page 94](#)
- [“Configuration de pool de stockage RAID-Z” à la page 94](#)
- [“Données d'autorétablissement dans une configuration redondante” à la page 96](#)
- [“Entrelacement dynamique dans un pool de stockage” à la page 96](#)

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 simple 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 97](#).

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 ou double. Une configuration RAID-Z à parité simple est similaire à une configuration RAID-5. Une configuration RAID-Z à double parité est similaire à une configuration RAID-6.

Tous les algorithmes similaires à RAID-5 traditionnels (RAID-4, RAID-6, RDP et EVEN-ODD, par exemple) présentent un problème connu appelé "RAID-5 write hole". 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 transmis au disque, la parité n'est pas synchronisée avec les données, et est par conséquent inutile à tout jamais (à moins qu'elle ne soit remplacée 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. Par exemple, si vous disposez de trois disques pour une configuration RAID-Z à parité simple, les données de parité occupent un espace 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
```

Une configuration RAID-Z conceptuelle plus complexe serait similaire à ce qui suit :

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

Si vous créez une configuration RAID-Z avec un grand nombre de disques, comme dans cet exemple, il est recommandé de scinder une configuration RAID-Z avec 14 disques en deux groupements de 7 disques. Les configurations RAID-Z disposant de groupements de moins de 10 disques devraient présenter de meilleures performances.

Pour plus d'informations sur la création de pools de stockage RAID-Z, consultez la section "Création de pools de stockage RAID-Z" à la page 98.

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, consultez le blog suivant :

http://blogs.sun.com/roller/page/roch?entry=when_to_and_not_to

Pour obtenir des informations supplémentaires sur les recommandations relatives aux pools de stockage RAID-Z, consultez le site des pratiques ZFS recommandées :

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

Données d'autorétablissement dans une configuration redondante

ZFS fournit des données d'autorétablissement dans une configuration RAID-Z ou 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

Pour chaque périphérique virtuel ajouté au pool, ZFS entrelace les données de façon dynamique sur l'ensemble des périphériques disponibles. 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 des périphériques virtuels sont ajoutés à un pool, ZFS attribue graduellement les données au nouveau périphérique afin de maintenir les performances et les politiques d'allocation d'espace. 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 assure un contrôle flexible des caractéristiques par défaut du pool. Par exemple, vous pouvez créer les configurations suivantes à partir de 4 disques :

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

ZFS assure la prise en charge de différents types de périphériques virtuels au sein du même pool, mais cette pratique est déconseillée. 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). Il est conseillé d'utiliser des périphériques virtuels de niveau supérieur du type identique et avec le même niveau de redondance dans tous les périphériques.

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 d'un pool de stockage ZFS” à la page 97
- “Gestion d'erreurs de création de pools de stockage ZFS” à la page 101
- “Destruction de pools de stockage ZFS” à la page 105
- “Affichage des informations d'un périphérique virtuel de pool de stockage” à la page 100

De par leur conception, la création et la destruction de pools est rapide et facile. Cependant, il se doit de réaliser ces opérations 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. La destruction d'un pool est encore plus facile. Utilisez la commande `zpool destroy` avec précaution. Cette commande simple a des conséquences significatives.

Création d'un pool 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 40.

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 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 mises en miroir recommandées, consultez le site suivant :

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

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

- Ajout d'un autre jeu de disques comme stockage vdev de niveau supérieur supplémentaire 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 106.](#)
- 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 109.](#)
- 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 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 114.](#)
- Séparation d'un ou de plusieurs disques dans une configuration existante, 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 109.](#)

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

- Vous ne pouvez pas supprimer totalement un périphérique d'un pool de stockage mis en miroir. Cette fonction fait l'objet d'une demande d'amélioration.
- Vous ne pouvez pas diviser ou fractionner un miroir à des fins de sauvegarde. Cette fonction fait l'objet d'une demande d'amélioration.

Création de pools 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 chemins complets. Le périphérique `/dev/dsk/c5t0d0` est identique au périphérique `c5t0d0`.

Il est possible de créer une configuration similaire avec des tranches de disque. Exemple :

```
# zpool create tank raidz c1t0d0s0 c2t0d0s0 c3t0d0s0 c4t0d0s0 c5t0d0s0
```

Cependant, vous devez préformater les disques afin de disposer d'une tranche zéro de taille adéquate.

Pour créer une configuration RAID-Z à double parité, appliquez le mot-clé `raidz2` lors de la création du pool. Exemple :

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

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

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

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

- Ajout d'un autre jeu de disques comme stockage vdev de niveau supérieur supplémentaire 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 106](#).
- 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 114](#).

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.
- Séparation d'un disque d'une configuration RAID-Z.
- Vous ne pouvez pas supprimer totalement un périphérique 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 94](#).

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

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. Pour plus d'informations sur les périphériques de journalisation ZFS, reportez-vous à la section [“Configuration de périphériques de journalisation ZFS distincts” à la page 21](#).

Vous pouvez configurer un périphérique de journalisation ZFS à la création du pool de stockage ou à un moment ultérieur.

Créez par exemple un pool de stockage mis en miroir à l'aide de périphériques de journalisation mis en miroir.

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

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

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

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

Chaque pool de stockage se compose d'un ou plusieurs périphériques virtuels. Un *périphérique virtuel* est une représentation interne du pool de stockage qui décrit la disposition du stockage physique et ses caractéristiques par défaut. 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 indéfini de périphériques virtuels à la racine de la configuration, appelés *périphériques virtuels racine*.

Deux périphériques virtuels racine, ou de niveau supérieur, assurent la redondance des données : les périphériques virtuels miroir et RAID-Z. Ces périphériques virtuels se composent de disques, de tranches de disques ou de fichiers. Un périphérique de rechange est un périphérique virtuel particulier qui maintient la liste des hot spare disponibles d'un pool.

L'exemple suivant illustre la création d'un pool composé de deux périphériques virtuels racine, 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 racine de 4 disques.

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

Vous pouvez ajouter un autre périphérique virtuel racine à ce pool à l'aide de la commande `zpool add`. Exemple :

```
# zpool add mypool raidz2 c2d0 c3d0 c4d0 c5d0
```

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`. Exemple :

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

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

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 d'entre elles peuvent être ignorées à l'aide de l'option -f, mais pas toutes. les utilisations 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.

Élément 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 soit pas monté et n'apparaisse pas comme étant en cours d'utilisation.

Élément d'un volume

Le disque fait partie d'un volume SVM.

Live upgrade

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

Élément 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`.

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
```

L'option `-f` permet d'ignorer ces erreurs, mais 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. Cette configuration est autorisée mais, si les niveaux de redondance sont incohérents, une partie de l'espace du périphérique de plus grande taille reste inutilisée. L'option `-f` est alors requise pour ignorer l'avertissement.

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

La création d'un pool peut échouer de manière inopinée, de différentes façons. Comme le formatage de disques constitue une action potentiellement néfaste, la commande `zpool create` dispose d'une option supplémentaire, `-n`, qui simule la création d'un pool sans écriture réelle sur le périphérique. Cette option 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 sans écriture des données. La commande `create -n` peut indiquer que l'opération s'effectue correctement, alors que la création du pool lors de son exécution effective échoue.

Point de montage par défaut pour les pools de stockage

Lors de la création d'un pool, le point de montage par défaut du jeu de données racine est */nom-pool*. Le répertoire doit être inexistant ou vide. Le répertoire est créé automatiquement s'il n'existe pas. Si le répertoire est vide, le jeu de données racine 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` :

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

# 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 un pool `home` et le jeu de données `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 164](#).

Destruction de pools de stockage ZFS

La commande `zpool destroy tank` 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 135.

Destruction d'un pool avec des périphériques défaillants

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 endommagés.

Ces périphériques, lorsqu'ils sont correctement réparés, sont signalés comme étant *potentiellement actifs* lors de la création d'un pool et s'affichent en tant que périphériques valides lorsque vous recherchez des pools à importer. Si un pool a tant de périphérique défaillant que le pool lui-même est défaillant (en d'autres termes, un périphérique virtuel de niveau supérieur est défaillant), alors 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 ou non. 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 127.

Pour de plus amples informations sur l'importation de pools, reportez-vous à la section [“Importation de pools de stockage ZFS”](#) à la page 134.

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 91. Une fois la création d'un pool terminée, 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 106
- “Connexion et séparation de périphériques dans un pool de stockage” à la page 109
- “Mise en ligne et mise hors ligne de périphériques dans un pool de stockage” à la page 111
- “Suppression des périphériques de pool de stockage” à la page 114
- “Remplacement de périphériques dans un pool de stockage” à la page 114
- “Désignation des disques hot spare dans le pool de stockage” à la page 116

Ajout de périphériques à un pool de stockage

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

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

Le format de spécification des périphériques virtuels est le même que pour la commande `zpool create` et obéit aux mêmes règles. 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. 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 mis en miroir `c3t1d0` et `c3t2d0` à la configuration existante du fichier `zeepool`.

Pour plus d'informations sur la validation des périphériques virtuels, reportez-vous à la section “Détection des périphériques utilisés” à la page 102.

EXEMPLE 5-1 Ajout de disques à une configuration ZFS mise en miroir

Dans l'exemple suivant, un autre miroir est ajouté à la configuration ZFS mise en miroir existante sur un système Sun Fire x4500.

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

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

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

EXEMPLE 5-2 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 3 disques en pool de stockage avec deux périphériques RAID-Z composé de 3 disques.

```
# zpool status
pool: rpool
state: ONLINE
scrub: none requested
config:
  NAME          STATE          READ WRITE CKSUM
  rpool         ONLINE         0     0     0
    raidz1      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 rpool raidz c2t2d0 c2t3d0 c2t4d0
```

```
# zpool status
pool: rpool
state: ONLINE
scrub: none requested
config:
  NAME          STATE          READ WRITE CKSUM
  rpool         ONLINE         0     0     0
    raidz1      ONLINE         0     0     0
      c1t2d0    ONLINE         0     0     0
      c1t3d0    ONLINE         0     0     0
      c1t4d0    ONLINE         0     0     0
    raidz1      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 5-3 Ajout d'un périphérique de journalisation mis en miroir à un pool de stockage ZFS

L'exemple suivant illustre l'ajout d'un périphérique de journalisation mis en miroir à un pool de stockage mis en miroir. Pour plus d'informations sur l'utilisation de périphériques de journalisation dans votre pool de stockage, reportez-vous à la section [“Configuration de périphériques de journalisation ZFS distincts”](#) à la page 21.

EXEMPLE 5-3 Ajout d'un périphérique de journalisation mis en miroir à un pool de stockage ZFS
(Suite)

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

    NAME          STATE          READ WRITE CKSUM
    newpool       ONLINE         0     0     0
      mirror     ONLINE         0     0     0
        c1t9d0    ONLINE         0     0     0
        c1t10d0   ONLINE         0     0     0

errors: No known data errors
# zpool add newpool log mirror c1t11d0 c1t12d0
# zpool status newpool
pool: newpool
state: ONLINE
scrub: none requested
config:

    NAME          STATE          READ WRITE CKSUM
    newpool       ONLINE         0     0     0
      mirror     ONLINE         0     0     0
        c1t9d0    ONLINE         0     0     0
        c1t10d0   ONLINE         0     0     0
    logs         ONLINE         0     0     0
      mirror     ONLINE         0     0     0
        c1t11d0   ONLINE         0     0     0
        c1t12d0   ONLINE         0     0     0

errors: No known data errors
```

connexion d'un périphérique de journal à un périphérique journal 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.

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, mis en miroir ou non.

EXEMPLE 5-4 Conversion d'un pool de stockage bidirectionnel mis en miroir en un pool de stockage tridirectionnel mis en miroir

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

```
# zpool status
pool: zpool
state: ONLINE
scrub: none requested
config:
    NAME          STATE      READ WRITE CKSUM
    zpool         ONLINE    0     0     0
      mirror     ONLINE    0     0     0
        c0t1d0   ONLINE    0     0     0
        c1t1d0   ONLINE    0     0     0
errors: No known data errors
# zpool attach zpool c1t1d0 c2t1d0
# zpool status
pool: zpool
state: ONLINE
scrub: resilver completed after 0h2m with 0 errors on Thu Aug 28 09:50:11 2008
config:
    NAME          STATE      READ WRITE CKSUM
    zpool         ONLINE    0     0     0
      mirror     ONLINE    0     0     0
        c0t1d0   ONLINE    0     0     0
        c1t1d0   ONLINE    0     0     0
        c2t1d0   ONLINE    0     0     0
```

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

EXEMPLE 5-5 Conversion d'un pool de stockage ZFS non redondant en un pool de stockage ZFS mis en miroir

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

```
# zpool create tank c0t1d0
# zpool status
pool: tank
state: ONLINE
scrub: none requested
```

EXEMPLE 5-5 Conversion d'un pool de stockage ZFS non redondant en un pool de stockage ZFS mis en miroir (Suite)

```
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
  pool: tank
  state: ONLINE
  scrub: resilver completed after 0h2m with 0 errors on Thu Aug 28 09:54:11 2008
config:
  NAME          STATE      READ WRITE CKSUM
  tank          ONLINE    0    0    0
  mirror       ONLINE    0    0    0
  c0t1d0       ONLINE    0    0    0
  c1t1d0       ONLINE    0    0    0
```

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

```
# zpool detach zeepool c2t1d0
```

Cependant, cette opération est refusée en l'absence de réplique valide des données. Exemple :

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

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, il est possible d'indiquer à ZFS d'ignorer le périphérique en le mettant hors ligne. ZFS n'envoie aucune requête aux périphériques hors ligne.

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

Vous pouvez utiliser la commande `offline` pour déconnecter temporairement le stockage. Par exemple, si vous devez déconnecter physiquement une baie d'un jeu de commutateurs Fibre Channel et la connecter à un autre jeu, vous pouvez mettre les LUN hors ligne dans la baie utilisée dans les pools de stockage ZFS. Une fois la baie reconnectée et opérationnelle sur le nouveau jeu de commutateurs, vous pouvez mettre les mêmes LUN en ligne. La réargenture des données ajoutées aux pools de stockage alors que les LUN étaient hors ligne s'effectue sur les LUN, une fois ceux-ci en ligne.

Ce scénario est possible si les systèmes en question ont accès au stockage une fois qu'il est connecté aux nouveaux commutateurs, éventuellement par le biais de contrôleurs différents, et si les pools sont définis en tant que configurations RAID-Z ou en miroir.

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. Exemple :

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

Gardez les points suivants à l'esprit lors de la mise hors ligne de périphériques :

- Vous ne pouvez pas mettre un périphérique hors ligne au point où il devient défaillant. Vous ne pouvez par exemple pas mettre hors ligne deux périphériques d'une configuration RAID-Z ni ne pouvez mettre hors ligne un périphérique virtuel de niveau supérieur.

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

- Par défaut, l'état hors ligne est persistant. Le périphérique reste hors ligne lors du redémarrage du système.

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

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

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 114.](#)

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 123.](#)

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 127.](#)

Mise en ligne d'un périphérique

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

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

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 utiliser la mise en ligne d'un périphérique pour remplacer un disque. Si vous mettez un périphérique hors ligne, remplacez le lecteur puis tentez de le mettre en ligne, son état continue à indiquer qu'il est défaillant.

Si vous tentez de mettre un périphérique défaillant en ligne, un message similaire au suivant s'affiche à partir de la commande `cmd` :

```
# zpool online tank c1t0d0
Bringing device c1t0d0 online
#
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Thu Apr 24 03:07:05 MDT 2008
PLATFORM: SUNW,Sun-Fire-880, CSN: -, HOSTNAME: neo2
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: b8ed8b2b-2c22-4281-bbfa-dd92d3cd924d
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 [“Réparation d'un périphérique manquant”](#) à la page 257.

Suppression des périphériques 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. Exemple :

```
# zpool clear tank
```

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

```
# zpool clear tank c1t0d0
```

Pour de plus amples informations sur l'effacement d'erreurs de `zpool` reportez-vous à la section [“Suppression des erreurs transitoires”](#) à la page 260.

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. ZFS reconnaît qu'il s'agit d'un disque différent situé au même emplacement. Par exemple, pour remplacer un disque défaillant (`c1t1d0`), supprimez-le, puis ajoutez le disque de rechange au même emplacement à l'aide d'une syntaxe similaire à la suivante :

```
# zpool replace tank c1t1d0
```

Si vous remplacez le périphérique unique d'un pool de stockage non redondant, vous devez indiquer les deux périphériques. Exemple :

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

Les étapes de base de remplacement d'un disque sont les suivantes D :

- Le cas échéant, mettez le disque hors ligne à l'aide de la commande `zpool offline`.
- Enlevez le disque à remplacer.
- Insérez le disque de remplacement.
- Exécutez la commande `zpool replace`. Exemple :

```
# zpool replace tank c1t1d0
```

- Remettez le disque en ligne à l'aide de la commande `zpool online`.

Sur certains systèmes, notamment sur Sun Fire x4500, vous devez annuler la configuration d'un disque avant de le mettre hors ligne. Si vous remplacez simplement un disque par un autre dans le même logement de ce système, vous pouvez exécuter la commande `zpool replace` comme indiqué précédemment.

L'[Exemple 11-1](#) illustre le remplacement d'un disque sur ce système.

Gardez à l'esprit les considérations suivantes lorsque vous remplacez des périphériques dans un pool de stockage ZFS :

- Si vous activez la propriété de `pool autoreplace` (valeur "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 disponible sur tous les types de matériel.
- Le périphérique de remplacement doit être d'une taille supérieure ou égale à la taille minimale de tous les périphériques dans une configuration en miroir ou RAID-Z.
- Si la taille du périphérique de remplacement est supérieure, la capacité du pool s'accroît une fois le remplacement terminé. Actuellement, vous devez exporter et importer le pool pour connaître la capacité étendue. Exemple :

```
# zpool list tank
NAME  SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
tank 16.8G   94K  16.7G    0%  ONLINE  -
# zpool replace tank c0t0d0 c0t4d0
# zpool list tank
NAME  SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
tank 16.8G  112K  16.7G    0%  ONLINE  -
# zpool export tank
# zpool import tank
# zpool list tank
NAME  SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
tank 33.9G  114K  33.9G    0%  ONLINE  -
```

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 130.

- À l'heure actuelle, pour accroître la taille d'un volume LUN existant appartenant à un pool de stockage, vous devez également effectuer les étapes d'exportation et d'importation afin d'afficher la capacité de disque étendue.
- Le remplacement des nombreux 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 afin de garantir le fonctionnement des périphériques de remplacement et l'exactitude des données écrites.

Pour plus d'informations sur le remplacement de périphériques, reportez-vous à la section “Réparation d'un périphérique manquant” à la page 257 et à la section “Réparation d'un périphérique endommagé” à la page 259.

Désignation des disques hot spare dans le pool de stockage

La fonction de disque hot spare permet d'identifier les disques utilisables pour remplacer un périphérique défaillant dans un ou plusieurs pools 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` ;
- les périphériques hot spare peuvent être partagés entre plusieurs pools.

Désignez les périphériques en tant que disques hot spare une fois le pool créé. Exemple :

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

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

Désignez les disques hot spare en les ajoutant à un pool après la création de ce dernier.
Exemple :

```
# zpool add zeepool spare c1t3d0 c2t3d0
# zpool status zeepool
pool: zeepool
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
zeepool	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
c2t1d0	ONLINE	0	0	0
spares				
c1t3d0	AVAIL			
c2t3d0	AVAIL			

Plusieurs pools peuvent partager des périphériques désignés en tant que disque hot spare.
Exemple :

```
# zpool create zeepool mirror c1t1d0 c2t1d0 spare c1t2d0 c2t2d0
# zpool create tank raidz c3t1d0 c4t1d0 spare c1t2d0 c2t2d0
```

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

```
# zpool remove zeepool c1t2d0
# zpool status zeepool
pool: zeepool
state: ONLINE
scrub: none requested
config:
```

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

Un disque hot spare ne peut pas être supprimé s'il est en cours d'utilisation par le pool de stockage.

Lorsque vous utilisez des disques hot spare ZFS, gardez à l'esprit les points suivants :

- Actuellement, la commande `zpool remove` ne peut être utilisée que pour la suppression de disques hot spare.
- Ajoutez un disque de rechange d'une taille égale ou supérieure à celle du disque le plus volumineux au sein 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
```

- Un disque hot spare peut être partagé entre plusieurs pools. Toutefois, vous ne pouvez pas exporter un pool contenant un disque spare partagé utilisé à moins de vous servir de l'option `zpool export -f` (force). Ce comportement empêche l'éventuelle corruption de données lors de l'exportation d'un pool contenant un disque spare partagé utilisé alors qu'un autre pool tente d'utiliser le disque spare partagé du pool exporté. Lorsque vous exportez, à l'aide de l'option `-f`, un pool contenant un disque spare partagé utilisé, sachez que l'opération risque d'entraîner une corruption des données si un autre pool tente d'activer le disque spare partagé utilisé.

Activation et désactivation de disque 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 avec la commande `zpool replace`.
- Remplacement automatique – En cas de réception 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 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 l'original mais remplacez le périphérique à l'aide de la commande `zpool detach` pour déconnecter le disque spare. Si vous activez la propriété `autoreplace` (valeur "on"), le disque spare est automatiquement déconnecté du pool de disques spare lorsque le nouveau périphérique est inséré et que l'opération en ligne s'achève.

La commande `zpool replace` permet de remplacer un périphérique manuellement par un disque hot spare. Exemple :

```
# zpool replace zeepool c2t1d0 c2t3d0
# zpool status zeepool
```

```

pool: zeepool
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Thu Aug 28 09:41:49 2008
config:

```

NAME	STATE	READ	WRITE	CKSUM	
zeepool	ONLINE	0	0	0	
mirror	ONLINE	0	0	0	
c1t2d0	ONLINE	0	0	0	
spare	ONLINE	0	0	0	
c2t1d0	ONLINE	0	0	0	
c2t3d0	ONLINE	0	0	0	
spares					
c1t3d0	AVAIL				
c2t3d0	INUSE				currently in use

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

Tout périphérique défaillant est remplacé automatiquement si un disque hot spare est disponible. Exemple :

```

# zpool status -x
pool: zeepool
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-D3
scrub: resilver completed after 0h12m with 0 errors on Thu Aug 28 09:29:43 2008
config:

```

NAME	STATE	READ	WRITE	CKSUM	
zeepool	DEGRADED	0	0	0	
mirror	DEGRADED	0	0	0	
c1t2d0	ONLINE	0	0	0	
spare	DEGRADED	0	0	0	
c2t1d0	UNAVAIL	0	0	0	cannot open
c2t3d0	ONLINE	0	0	0	
spares					
c1t3d0	AVAIL				
c2t3d0	INUSE				currently in use

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

Il existe actuellement trois façons de désactiver les disques hot spare disponibles :

- annuler le disque hot spare en le supprimant du pool de stockage ;
- remplacer le périphérique d'origine par un disque hot spare ;

- échanger le disque hot spare de façon permanente.

Une fois le périphérique défaillant remplacé, utilisez la commande `zpool detach` pour retourner le disque hot spare au jeu restant. Exemple :

```
# zpool detach zeepool c2t3d0
# zpool status zeepool
  pool: zeepool
  state: ONLINE
  scrub: resilver completed with 0 errors on Mon Sep 22 14:23:06 2008
  config:
```

NAME	STATE	READ	WRITE	CKSUM
zeepool	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c1t2d0	ONLINE	0	0	0
c2t1d0	ONLINE	0	0	0
spares				
c1t3d0	AVAIL			
c2t3d0	AVAIL			

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

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. Exemple :

```
# zpool get all mpool
NAME  PROPERTY  VALUE                SOURCE
mpool size    33.8G              -
mpool used   5.91G             -
mpool available 27.8G            -
mpool capacity 17%               -
mpool altroot -                  default
mpool health  ONLINE            -
mpool guid    2689713858991441653 -
mpool version 10                 default
mpool bootfs  mpool/ROOT/zfs2BE local
mpool delegation on                 default
mpool autoreplace on                 local
mpool cachefile -                  default
mpool failmode continue          local
```

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

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

TABLEAU 5-1 Description des propriétés d'un pool ZFS

Nom de la propriété	Type	Valeur par défaut	Description
<code>altroot</code>	Chaîne	<code>off</code>	Identifie un répertoire racine 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 vous ne pouvez pas faire confiance aux points de montage ou dans un environnement d'initialisation alternatif dans lequel les chemins types sont incorrects.
<code>available</code>	Valeur numérique	SO	Valeur en lecture seule identifiant la quantité de stockage disponible au sein du pool. Cette propriété peut également s'afficher sous la forme du nom de colonne contracté, <code>avail</code> .
<code>autoreplace</code>	Booléen	<code>off</code>	Contrôle le remplacement automatique d'un périphérique. Si la propriété est désactivée, l'administrateur doit initier le remplacement du périphérique à l'aide de la commande <code>zpool replace</code> . Si la propriété est activée, tout nouveau périphérique se trouvant au même emplacement physique qu'un périphérique qui appartenait au pool est automatiquement formaté et remplacé. Le comportement par défaut est "off". Cette propriété peut également s'afficher sous la forme du nom de colonne contracté, <code>replace</code> .
<code>bootfs</code>	Booléen	SO	Identifie le jeu de données d'initialisation par défaut du pool racine. Cette propriété est censée être définie par les programmes d'installation et de mise à niveau.
<code>capacity</code>	Valeur numérique	SO	Valeur en lecture seule identifiant le pourcentage d'espace utilisé du pool. Cette propriété peut également s'afficher sous la forme du nom de colonne contracté, <code>cap</code> .
<code>delegation</code>	Booléen	<code>on</code>	Contrôle l'octroi des droits d'accès définis pour le jeu de données à un utilisateur sans privilège. Pour plus d'informations, reportez-vous au Chapitre9, "Administration déléguée de ZFS" .

TABLEAU 5-1 Description des propriétés d'un pool ZFS (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
<code>failmode</code>	Chaîne	<code>wait</code>	<p>Contrôle le comportement du système en cas de panne catastrophique du 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 tel événement est déterminé par une des valeurs suivantes :</p> <ul style="list-style-type: none"> ■ <code>wait</code> : bloque tout accès d'E/S jusqu'au rétablissement de la connectivité des périphériques et jusqu'à l'effacement des erreurs à l'aide de la commande <code>zpool clear</code>. Il s'agit du comportement par défaut. ■ <code>continue</code> : renvoie une erreur EIO à toute nouvelle requête d'E/S d'écriture mais autorise les lectures de tout autre périphérique fonctionnel. Toute requête 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 <code>zpool clear</code>. ■ <code>panic</code> : imprime un message sur la console et génère un vidage mémoire sur incident du système.
<code>guid</code>	Chaîne	SO	Propriété en lecture seule identifiant l'identificateur unique du pool.
<code>health</code>	Chaîne	SO	Propriété en lecture seule indiquant la fonctionnalité actuelle du pool ; les valeurs possibles sont : ONLINE, DEGRADED, FAULTED, OFFLINE, REMOVED ou UNAVAIL.
<code>size</code>	Valeur numérique	SO	Propriété en lecture seule identifiant la taille totale du pool de stockage.
<code>used</code>	Valeur numérique	SO	Propriété en lecture seule identifiant la quantité de stockage disponible au sein du pool.
<code>version</code>	Valeur numérique	SO	Identifie la version actuelle sur disque du pool. La valeur de cette propriété peut être augmentée mais ne peut jamais être diminuée. La méthode recommandée de mise à jour des pools consiste à utiliser la commande <code>zpool upgrade</code> , 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 <code>zpool upgrade -v</code> . La valeur <code>current</code> est un alias de la toute dernière version prise en charge.

Requête d'état de pool de stockage ZFS

La commande `zpool list` offre plusieurs moyens d'effectuer des requêtes 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 de pools de stockage ZFS de base” à la page 123](#)
- [“Visualisation de statistiques d'E/S de pools de stockage ZFS” à la page 125](#)
- [“Détermination de l'état de maintenance des pools de stockage ZFS” à la page 127](#)

Affichage des informations de pools de stockage ZFS de base

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

Liste des informations relatives aux pools de stockage

Sans arguments, la commande affiche tous les champs pour tous les pools dans le système.

Exemple :

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

La sortie 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.
USED	Quantité d'espace utilisée, c'est-à-dire allouée par tous les jeux de données et métadonnées internes. Notez que cette quantité d'espace 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 ZFS” à la page 50 .
AVAILABLE	Quantité d'espace disponible, c'est-à-dire non allouée dans le pool.
CAPACITY (CAP)	Capacité, ou quantité d'espace utilisée, exprimée en tant que pourcentage d'espace total.
HEALTH	État 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 127.

ALROOT

Racine de remplacement, le cas échéant.

Pour de plus amples informations sur les pools racine de remplacement, reportez-vous à la section [“Utilisation de pools racine ZFS de remplacement”](#) à la page 244.

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

```
# zpool list tank
NAME                SIZE    USED    AVAIL    CAP  HEALTH    ALROOT
tank                80.0G   22.3G   47.7G   28%  ONLINE   -
```

Liste de statistiques spécifiques à un pool de stockage

L'option `-o` permet d'effectuer une requête 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, utilisez la syntaxe suivante :

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

Les noms de colonnes correspondent aux propriétés répertoriées à la section [“Liste des informations relatives aux pools de stockage”](#) à la page 123.

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 requête suivante permet d'obtenir la liste des noms de pool dans le système :

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

Voici un autre exemple :

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

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

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

USED CAPACITY	Capacité utilisée, c'est-à-dire quantité de données actuellement stockées dans le pool ou le périphérique. Ce chiffre diffère quelque peu de la quantité d'espace 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 ZFS” à la page 50 .
AVAILABLE CAPACITY	Capacité disponible, c'est-à-dire quantité d'espace disponible dans le pool ou le périphérique. Comme pour la capacité utilisée, cette quantité diffère légèrement de la quantité d'espace 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 requêtes 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 de statistiques relatives à l'ensemble du pool

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

```
# zpool iostat
          capacity    operations    bandwidth
pool      used avail  read  write  read  write
```

```

-----
tank      100G 20.0G 1.2M 102K 1.2M 3.45K
dozer     12.3G 67.7G 132K 15.2K 32.1K 1.20K

```

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

```

# zpool iostat tank 2
          capacity      operations      bandwidth
pool     used  avail   read  write  read  write
-----
tank     100G 20.0G  1.2M  102K  1.2M  3.45K
tank     100G 20.0G    134     0  1.34K     0
tank     100G 20.0G    94   342  1.06K  4.1M

```

Dans cet exemple, la commande affiche les statistiques d'utilisation pour le pool tank uniquement, toutes les deux secondes, jusqu'à ce que vous saisissez Ctrl-C. Vous pouvez également spécifier un paramètre count supplémentaires pour entraîner l'interruption de la commande une fois le nombre spécifié d'itérations effectuées. Par exemple, `zpool iostat 2 3` imprimerait un résumé toutes les deux secondes pour trois itérations, pendant six secondes. S'il y a 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 de statistiques de périphériques virtuels

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

```

# zpool iostat -v
          capacity      operations      bandwidth
tank     used  avail   read  write  read  write
-----
mirror   20.4G 59.6G     0    22     0  6.00K
  c1t0d0     -     -     1   295  11.2K  148K
  c1t1d0     -     -     1   299  11.2K  148K
-----
total    24.5K 149M     0    22     0  6.00K

```

Notez deux points importants lors de l'affichage de statistiques d'E/S par périphérique virtuel :

- Tout d'abord, l'utilisation d'espace n'est disponible que pour les périphériques virtuels de niveau supérieur. L'allocation d'espace entre les périphériques virtuels RAID-Z et les miroirs 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 en tant que partie de création de pool qui n'est pas comptabilisée au niveau du miroir. Avec le temps, ces chiffres devraient s'égaliser graduellement, mais les périphériques défectueux, ne répondant pas ou mis hors ligne 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 `fmfd` et s'affichent dans la console système et 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 11, "Résolution de problèmes et récupération de données ZFS"](#).

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

ONLINE	Le périphérique est en état de fonctionnement normal. Bien que certaines erreurs transitoires puissent se produire, le périphérique est en état de bon fonctionnement.
DEGRADED	Le périphérique virtuel a subi une défaillance mais est toujours capable 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.
FAULTED	Le périphérique virtuel est totalement inaccessible. Cet état indique en règle générale une défaillance totale du périphérique, de telle façon que ZFS est

incapable d'y envoyer des données ou d'en recevoir de lui. Si un périphérique virtuel de niveau supérieur se trouve dans cet état, le pool est totalement inaccessible.

OFFLINE	Le périphérique virtuel a été mis hors ligne explicitement par l'administrateur.
UNAVAILABLE	L'ouverture du périphérique ou du périphérique virtuel est impossible. Dans certains cas, les pools avec des périphériques en état UNAVAILABLE s'affichent en mode DEGRADED. Si un périphérique de niveau supérieur est indisponible, aucun élément du pool n'est accessible.
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 plate-formes.

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 UNAVAILABLE, l'état du pool est également DEGRADED. Si l'état d'un des périphériques virtuels est FAULTED ou OFFLINE, l'état du pool est également FAULTED. Un pool en état défaillant est totalement inaccessible. Aucune donnée ne peut être récupérée tant que les périphériques nécessaires n'ont pas été connectés ou réparés. Un pool en état dégradé continue de s'exécuter, mais vous risquez de ne pas atteindre le même niveau de redondance de données ou de traitement de données que si le pool était en ligne.

État de maintenance de base de pool de stockage

La commande `zpool status` constitue la manière la plus simple d'effectuer une requête relative à une vue d'ensemble de l'état de maintenance d'un pool :

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

Il est possible d'examiner des pools spécifiques en spécifiant un nom de pool dans la 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.

État de maintenance détaillé

L'option `-v` permet d'effectuer une requête pour obtenir un résumé détaillé de l'état de maintenance. 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: none requested
config:

```

NAME	STATE	READ	WRITE	CKSUM	
tank	DEGRADED	0	0	0	
mirror	DEGRADED	0	0	0	
c1t0d0	FAULTED	0	0	0	cannot open
c1t1d0	ONLINE	0	0	0	

```
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 ci-dessus, le périphérique défaillant devrait être remplacé. Une fois le périphérique remplacé, exécutez la commande `zpool online` pour le remettre en ligne. Exemple :

```

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

```

Si un périphérique d'un pool est hors ligne, la sortie de commande identifie le pool qui pose problème. 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: none requested
config:

```

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

```
clt1d0  OFFLINE      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 qui se sont produites sur le périphérique. Il est probable que ces erreurs correspondent à une défaillance potentielle de périphérique et que des actions correctives soient 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 nombre d'erreurs identifie toute erreur de données connue.

Dans l'exemple de sortie ci-dessus, 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 défaillants, reportez-vous au [Chapitre 11, “Résolution de problèmes et récupération de données ZFS”](#).

Migration de pools de stockage ZFS

Parfois, il est possible que vous deviez déplacer un pool de stockage d'une machine à l'autre. Pour ce faire, les périphériques de stockage doivent être déconnectés de la machine d'origine et reconnectés à la machine de destination. Pour accomplir cette tâche, vous pouvez raccorder physiquement les périphériques ou utiliser des périphériques multiport, par exemple les périphériques d'un SAN. ZFS permet d'exporter le pool à partir d'une machine et de l'importer vers la machine de destination, même si les endianness des deux machines sont différents. 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 machines différentes, reportez-vous à la section “Envoi et réception de données ZFS” à la page 184.

- “Préparatifs de migration de pool de stockage ZFS” à la page 130
- “Exportation d'un pool de stockage ZFS” à la page 131
- “Définition des pools de stockage disponibles pour importation” à la page 131
- “Recherche de pools de stockage ZFS dans d'autres répertoires” à la page 134
- “Importation de pools de stockage ZFS” à la page 134
- “Récupération de pools de stockage ZFS détruits” à la page 135
- “Mise à niveau de pools de stockage ZFS” à la page 137

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 connaissance du pool dans le 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 étant défaillant sur la machine d'origine dans la mesure où les périphériques ne sont plus présents. Par défaut, la machine 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 composé de stockage connecté au réseau toujours en cours d'utilisation dans un autre système.

Exportation d'un pool de stockage ZFS

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

```
# zpool export tank
```

Une fois la commande exécutée, le pool `tank` n'est plus visible sur le système. La commande tente de démonter tout système de fichiers démonté au sein du pool avant de continuer. Si le démontage d'un des système de fichiers est impossible, vous pouvez le forcer à l'aide de l'option `-f`. Exemple :

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

Si les périphériques ne sont pas disponibles lors de l'export, les disques ne peuvent pas être spécifiés 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 235.

Définition des pools de stockage disponibles pour importation

Une fois le pool supprimé du système (soit par le biais de l'exportation, soit par le biais d'une suppression forcée des périphériques), connectez les périphériques au système cible. Bien que ZFS soit capable de gérer des situations dans lesquelles seule une partie des périphériques est disponible, tous les périphériques dans le pool doivent être déplacés d'un système à l'autre. 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. Exemple :

```
# zpool import
pool: tank
   id: 3778921145927357706
  state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

      tank      ONLINE
      mirror    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 identificateur numérique afin de les distinguer.

Tout comme la commande `zpool status`, la commande `zpool import` se rapporte à un article de connaissances disponible sur le Web avec 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 corruption 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 la redondance est suffisante pour obtenir un pool utilisable, le pool s'affiche dans l'état `DEGRADED`. Exemple :

```
# zpool import
pool: tank
   id: 3778921145927357706
  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:

      tank      DEGRADED
      mirror    DEGRADED
         c1t0d0  UNAVAIL  cannot open
         c1t1d0  ONLINE
```

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 défectueux ou manquants est trop importants, l'importation du pool est impossible. Exemple :

```
# zpool import
pool: dozer
  id: 12090808386336829175
  state: FAULTED
action: The pool cannot be imported. Attach the missing
       devices and try again.
  see: http://www.sun.com/msg/ZFS-8000-6X
config:
  raidz          FAULTED
    c1t0d0      ONLINE
    c1t1d0      FAULTED
    c1t2d0      ONLINE
    c1t3d0      FAULTED
```

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 les autres périphériques faisant partie du pool, mais fournit autant d'informations que possible sur la situation. Exemple :

```
# zpool import
pool: dozer
  id: 12090808386336829175
  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
    raidz        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.
```

Recherche de pools de stockage ZFS dans 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 différents répertoires. Exemple :

```
# zpool create dozer mirror /file/a /file/b
# zpool export dozer
# zpool import -d /file
  pool: dozer
    id: 1095241472586793582
   state: ONLINE
 action: The pool can be imported using its name or numeric identifier.
config:

        dozer      ONLINE
         mirror    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 importation, vous pouvez l'importer en spécifiant son nom ou son identifiant numérique en tant qu'argument pour la commande `zpool import`. Exemple :

```
# zpool import tank
```

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

```
# zpool import dozer zeepool
```

Cette commande importe le pool dozer exporté sous le nouveau nom zeepool. 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.

Exemple :

```

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

```

Les pools peuvent également être importés sous une racine de remplacement à l'aide de l'option `-R`. Pour plus d'informations sur les pools racine de remplacement, reportez-vous à la section [“Utilisation de pools racine ZFS de remplacement” à la page 244](#).

Récupération de pools de stockage ZFS détruits

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

```

# zpool destroy tank
# zpool import -D
pool: tank
  id: 3778921145927357706
  state: ONLINE (DESTROYED)
action: The pool can be imported using its name or numeric identifier. The
        pool was destroyed, but can be imported using the '-Df' flags.

```

config:

```
tank      ONLINE
mirror    ONLINE
c1t0d0    ONLINE
c1t1d0    ONLINE
```

Dans la sortie de `zpool import`, vous pouvez identifier ce pool 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 et l'option `-f`. Exemple :

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

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c1t0d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0

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

Même si l'un des périphériques du pool détruit est défaillant ou indisponible, vous devriez être en mesure de récupérer le pool détruit. Dans ce cas, importez le pool défaillant et tentez ensuite de réparer la défaillance du périphérique. Exemple :

```
# zpool destroy dozer
# zpool import -D
pool: dozer
id:
state: DEGRADED (DESTROYED)
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. The
        pool was destroyed, but can be imported using the '-Df' flags.
see: http://www.sun.com/msg/ZFS-8000-2Q
config:
```

```
dozer      DEGRADED
```

```

raidz      ONLINE
c1t0d0    ONLINE
c1t1d0    ONLINE
c1t2d0    UNAVAIL  cannot open
c1t3d0    ONLINE
# 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-D3
scrub: resilver completed after 0h0m with 0 errors on Thu Aug 28 10:01:48 2008
config:

```

NAME	STATE	READ	WRITE	CKSUM
dozer	DEGRADED	0	0	0
raidz	ONLINE	0	0	0
c1t0d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
c1t2d0	UNAVAIL	0	0	0 cannot open
c1t3d0	ONLINE	0	0	0

```

errors: No known data errors
# zpool online dozer c1t2d0
Bringing device c1t2d0 online
# 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 Solaris précédente (la version Solaris 10 6/06, par exemple), vous pouvez mettre les pools à niveau à l'aide de la commande `zpool upgrade` pour bénéficier des fonctions des pools de la version Solaris 10 11/06. De plus, la commande `zpool status` indique dorénavant si la version actuelle des pools est plus ancienne. Exemple :

```

# zpool status
pool: test
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
test	ONLINE	0	0	0
c1t27d0	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 version 3.
```

```
The following versions are supported:
```

```
VER  DESCRIPTION
---  -----
 1  Initial ZFS version
 2  Ditto blocks (replicated metadata)
 3  Hot spares and double parity RAID-Z
```

```
For more information on a particular version, including supported releases, see:
```

```
http://www.opensolaris.org/os/community/zfs/version/N
```

```
Where 'N' is the version number.
```

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

```
# zpool upgrade -a
```

Remarque – En cas de mise à niveau des pools vers la version la plus récente, il sera désormais impossible d'y accéder sur des systèmes exécutant des versions antérieures de ZFS.

Gestion des systèmes de fichiers ZFS

Ce chapitre contient des informations détaillées sur la gestion des systèmes de fichiers Solaris™ ZFS. 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.

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 7](#), "Utilisation des instantanés et des clones ZFS". Pour plus d'informations sur les volumes émulsés, reportez-vous à la section "Volumes ZFS" à la page 235.

Remarque – Dans ce chapitre, le terme *jeu de données* désigne de manière générique un système de fichiers, un instantané, un clone ou un volume.

Il contient les sections suivantes :

- "Création et destruction de systèmes de fichiers ZFS" à la page 140
- "Présentation des propriétés ZFS" à la page 143
- "Envoi de requêtes sur les informations des systèmes de fichiers ZFS" à la page 156
- "Gestion des propriétés ZFS" à la page 158
- "Montage et partage des systèmes de fichiers ZFS" à la page 164
- "Quotas et réservations ZFS" à la page 170

- [“Envoi et réception de données ZFS” à la page 184](#)

Création et destruction 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`.

- [“Création d'un système de fichiers ZFS” à la page 140](#)
- [“Destruction d'un système de fichiers ZFS” à la page 141](#)
- [“Modification du nom d'un système de fichiers ZFS” à la page 142](#)

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 :

nom-pool/[nom-système-fichiers/]nom-système-fichiers

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. Tous les noms des systèmes de fichiers intermédiaires doivent déjà être définis dans le pool. 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 40](#).

Dans l'exemple suivant, un système de fichiers nommé `bonwick` est créé dans le système de fichiers `tank/home`.

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

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 */jeu-données*, à l'aide du chemin défini pour le nom du système dans la commande `create`. Dans cet exemple, le fichier créé `bonwick` est monté sous `/tank/home/bonwick`. 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 164](#).

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 défini et 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 143.

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 165.

L'exemple suivant illustre la destruction du système de fichiers `tabriz`.

```
# zfs destroy tank/home/tabriz
```



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/ahrens
cannot unmount 'tank/home/ahrens': Device busy
```

```
# zfs destroy -f tank/home/ahrens
```

La commande `zfs destroy` échoue également si le système de fichiers possède des systèmes enfant. 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/billm
tank/ws/bonwick
tank/ws/maybee
```

```
# zfs destroy -r tank/ws
```

Si le système de fichiers à détruire possède des systèmes indirectement dépendants, la commande de destruction récursive mentionnée ci-dessus échoue. Pour forcer la destruction de *tous* 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/schrock
cannot destroy 'tank/home/schrock': filesystem has dependent clones
use '-R' to destroy the following datasets:
tank/clones/schrock-clone
```

```
# zfs destroy -R tank/home/schrock
```



Attention – Aucune invite de confirmation ne s'affiche lors de l'utilisation des options `-f`, `-r` ou `-R`. L'utilisation de ces options requiert donc une attention particulière.

Pour plus d'informations sur les instantanés et les clones, reportez-vous au [Chapitre 7](#), “Utilisation des instantanés et des clones ZFS”.

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`. Cette commande permet d'effectuer les opérations suivantes : Modification du nom

- modifier le nom d'un système de fichiers ;
- modifier l'emplacement d'un 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 ci-dessous illustre la modification du nom d'un système de fichiers à l'aide de la sous-commande `rename` :

```
# zfs rename tank/home/kustarz tank/home/kustarz_old
```

Cette commande entraîne le remplacement du nom du système de fichiers `kustarz` par `kustarz_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/maybee tank/ws/maybee
```

Dans cet exemple, le système de fichiers `maybee` 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 disponible doit être

suffisant pour contenir le nouveau système de fichiers. Si l'espace disponible du nouvel emplacement est insuffisant (par exemple, si le quota d'espace disque est atteint), l'opération échoue.

Pour plus d'informations sur les quotas, reportez-vous à la section [“Quotas et réservations ZFS” à la page 170](#).

L'opération de modification du nom tente de démonter, puis de remonter le système de fichiers ainsi que ses éventuels systèmes de fichiers descendants. Si la commande ne parvient pas à démonter un système de fichiers actif, l'opération échoue. Dans ce cas, 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 179](#).

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 la section s'appliquent à tous les types de jeu de données.

- [“Propriétés ZFS natives en lecture seule” à la page 150](#)
- [“Propriétés ZFS natives définies” à la page 151](#)
- [“Propriétés ZFS définies par l'utilisateur” à la page 154](#)

Les propriétés se divisent en deux catégories : les propriétés natives et les propriétés définies par l'utilisateur. Les propriétés natives permettent d'exporter des statistiques internes ou de contrôler le comportement des systèmes 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 154](#).

La plupart des propriétés pouvant être définies peuvent également être héritées. Les propriétés pouvant être héritées sont des propriétés qui, une fois définies sur un système parent, peuvent être appliquées à l'ensemble des descendants de ce parent.

Toutes ces propriétés sont associées à une source. Cette source indique la manière dont la propriété a été obtenue. Les sources de propriétés peuvent être définies sur les valeurs suivantes :

`local`

Une source définie sur la valeur `local` indique que la propriété a 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 158](#).

<code>inherited from <i>nom-jeu-données</i></code>	La valeur <code>inherited from <i>nom-jeu-données</i></code> signifie que la propriété a été héritée du système ascendant indiqué.
<code>default</code>	La valeur <code>default</code> signifie que le paramètre de la propriété n'a été ni hérité, ni défini de manière locale. Cette source est définie lorsque la propriété n'est associée à la source <code>local</code> 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 154.

TABLEAU 6-1 Description des propriétés ZFS natives

Nom de la propriété	Type	Valeur par défaut	Description
<code>aclinherit</code>	Chaîne	<code>secure</code>	Contrôle le processus d'héritage des entrées ACL lors de la création de fichiers et de répertoires. Les valeurs possibles sont <code>discard</code> , <code>noallow</code> , <code>secure</code> et <code>passthrough</code> . Pour une description de ces valeurs, reportez-vous à la section “ Modes de propriétés d'ACL ” à la page 197.
<code>aclmode</code>	Chaîne	<code>groupmask</code>	Contrôle le processus de modification des entrées ACL lors des opérations <code>chmod</code> . Les valeurs possibles sont <code>discard</code> , <code>groupmask</code> et <code>passthrough</code> . Pour une description de ces valeurs, reportez-vous à la section “ Modes de propriétés d'ACL ” à la page 197.
<code>atime</code>	Booléen	<code>on</code>	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.

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
available	Valeur numérique	SO	<p>Propriété en lecture seule indiquant la quantité d'espace disponible pour le jeu de données et l'ensemble des systèmes enfant, sans tenir compte des autres activités du pool. L'espace 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.</p> <p>Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>avail</code>.</p> <p>Pour plus d'informations sur la détermination de l'espace disque, reportez-vous à la section “Comptabilisation de l'espace ZFS” à la page 50.</p>
canmount	Booléen	on	<p>Détermine si le système de fichiers peut être monté à l'aide de la commande <code>zfs mount</code>. 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 <code>off</code>, 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.</p> <p>Lorsque l'option <code>noauto</code> est définie, un jeu de données ne peut être monté et démonté que de manière explicite. Le jeu de données n'est pas monté automatiquement lorsqu'il est créé ou importé, et n'est pas monté par la commande <code>zfs mount -a</code> ni démonté par la commande <code>zfs unmount -a</code>.</p> <p>Pour plus d'informations, reportez-vous à la section “Propriété canmount” à la page 153.</p>
Somme de contrôle	Chaîne	on	<p>Détermine la somme de contrôle permettant de vérifier l'intégrité des données. La valeur par défaut est définie sur <code>on</code>. Cette valeur permet de sélectionner automatiquement l'algorithme approprié, actuellement <code>fletcher2</code>. Les valeurs possibles sont <code>on</code>, <code>off</code>, <code>fletcher2</code>, <code>fletcher4</code> et <code>sha256</code>. La valeur <code>off</code> entraîne la désactivation du contrôle d'intégrité des données utilisateur. La valeur <code>off</code> n'est pas recommandée.</p>

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
<code>compression</code>	Chaîne	<code>off</code>	<p>Active ou désactive la compression de ce jeu de données. Les valeurs sont <code>on</code>, <code>off</code> et <code>lzjb</code>, <code>gzip</code> ou <code>gzip-N</code>. Donner à cette propriété la valeur <code>lzjb</code>, <code>gzip</code> ou la valeur <code>gzip-N</code> a actuellement le même effet que la valeur <code>on</code>. La valeur par défaut est <code>off</code>. 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.</p> <p>Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>compress</code>.</p>
<code>compressratio</code>	Valeur numérique	SO	<p>Propriété en lecture seule indiquant le ratio de compression obtenu pour le jeu de données, exprimé sous la forme d'un multiple. La compression peut être activée en exécutant <code>zfs set compression=on jeu de données</code>.</p> <p>Ce ratio est calculé sur la base de la taille logique de l'ensemble des fichiers et de la quantité de données physiques indiquée. La propriété induit un gain explicite basé sur l'utilisation de la propriété <code>compression</code>.</p>
<code>copies</code>	Valeur numérique	1	<p>Définit le nombre de copies des données utilisateur par système de fichiers. Les valeurs disponibles sont 1, 2 et 3. Ces copies viennent s'ajouter à toute redondance au niveau du pool. L'espace 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 <code>used</code> 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.</p>
<code>creation</code>	Chaîne	SO	<p>Propriété en lecture seule identifiant la date et l'heure de création de ce jeu de données.</p>
<code>devices</code>	Booléen	<code>on</code>	<p>Contrôle la capacité à ouvrir les fichiers des périphériques dans le système de fichiers.</p>
<code>exec</code>	Booléen	<code>on</code>	<p>Contrôle l'autorisation d'exécuter les programmes contenus dans le système de fichiers. Par ailleurs, lorsqu'elle est définie sur <code>off</code>, les appels de la commande <code>mmap(2)</code> avec <code>PROT_EXEC</code> ne sont pas autorisés.</p>

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
mounted	Booléen	SO	Propriété en lecture seule indiquant si le système de fichiers, le clone ou l'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 <code>mountpoint</code> d'un système de fichiers est modifiée, ce système de fichiers ainsi que les éventuels systèmes enfant héritant du point de montage sont démontés. Si la nouvelle valeur est définie sur <code>legacy</code> , ces systèmes restent démontés. Dans le cas contraire, ils sont automatiquement remontés au nouvel emplacement si la propriété était précédemment définie sur <code>legacy</code> ou sur <code>none</code> 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 164.
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 <code>-r</code> ou <code>-f</code>) tant que le clone existe. Les systèmes de fichiers non clonés n'indiquent aucune origine.
quota	Valeur numérique (ou none)	none	Limite la quantité d'espace disponible pour le jeu de données et ses descendants. Cette propriété permet d'appliquer une limite fixe à la quantité d'espace 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 d'un jeu de données 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 <code>volsize</code> 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 171.
readonly	Booléen	off	Contrôle l'autorisation de modifier le jeu de données. Lorsqu'elle est définie sur <code>on</code> , aucune modification ne peut être apportée au jeu de données. Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>rduonly</code> .

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
recordsize	Valeur numérique	128K	<p>Spécifie une taille de bloc suggérée pour les fichiers du système de fichiers.</p> <p>Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>recsize</code>. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété <code>recordsize</code>” à la page 153.</p>
referenced	Valeur numérique	SO	<p>Propriété en lecture seule identifiant la quantité de données à laquelle ce jeu de données a accès, lesquelles peuvent être ou non partagées avec d'autres jeux de données du pool.</p> <p>Lorsqu'un instantané ou un clone est créé, il indique dans un premier temps la même quantité d'espace que le système de fichiers ou l'instantané à partir duquel il a été créé. En effet, son contenu est identique.</p> <p>Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>refer</code>.</p>
refquota	Valeur numérique (ou none)	none	<p>Définit la quantité d'espace pouvant être utilisé par un jeu de données. Cette propriété définit une quantité d'espace maximale. Cette limite maximale n'inclut pas l'espace utilisé par les descendants, notamment les instantanés et les clones.</p>
refreservation	Valeur numérique (ou none)	none	<p>Définit la quantité d'espace minimale garantie pour un jeu de données, à l'exclusion des descendants, notamment les instantanés et les clones. Lorsque la quantité d'espace utilisée est inférieure à cette valeur, le système considère que le jeu de données utilise la quantité d'espace spécifiée par <code>refreservation</code>. La réservation <code>refreservation</code> est prise en compte dans l'espace utilisé des jeux de données parent et vient en déduction de leurs quotas et réservations.</p> <p>Lorsque la propriété <code>refreservation</code> 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 <i>référéncés</i> dans le jeu de données.</p> <p>Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>refreserv</code>.</p>

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
reservation	Valeur numérique (ou none)	none	<p>Quantité minimale d'espace réservée pour un jeu de données et ses descendants. Lorsque la quantité d'espace utilisée est inférieure à la valeur de cette propriété, le système considère que le jeu de données utilise la quantité d'espace réservée. Les réservations sont prises en compte dans l'espace utilisé des jeux de données parent et viennent en déduction de leurs quotas et réservations.</p> <p>Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>reserv</code>.</p> <p>Pour plus d'informations, reportez-vous à la section “Définition de réservations sur les systèmes de fichiers ZFS” à la page 173.</p>
setuid	Booléen	on	<p>Contrôle l'application du bit <code>setuid</code> dans le système de fichiers.</p>
sharenfs	Chaîne	off	<p>Détermine si le système de fichiers est disponible via NFS, ainsi que les options utilisées. Si cette propriété est définie sur <code>on</code>, la commande <code>zfs share</code> est exécutée sans option. Dans le cas contraire, la commande <code>zfs share</code> est exécutée avec les options équivalentes au contenu de cette propriété. Si elle est définie sur <code>off</code>, le système de fichiers est géré à l'aide des commandes héritées <code>share</code> et <code>unshare</code> et du fichier <code>dfstab</code> file.</p> <p>Pour plus d'informations sur le partage des systèmes de fichiers ZF, reportez-vous à la section “Activation et annulation du partage des systèmes de fichiers ZFS” à la page 169.</p>
snapdir	Chaîne	hidden	<p>Détermine si le répertoire <code>.zfs</code> doit être affiché ou masqué au niveau de la racine 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 177.</p>
type	Chaîne	SO	<p>Propriété en lecture seule identifiant le type de jeu de données comme étant un système de fichiers, (<code>filesystem</code>; système de fichiers à proprement parler ou clone), un volume (<code>volume</code>) ou un instantané (<code>snapshot</code>).</p>
used	Valeur numérique	SO	<p>Propriété en lecture seule identifiant la quantité d'espace utilisée par le jeu de données et tous ses descendants.</p> <p>Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété used” à la page 151.</p>

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
<code>volsize</code>	Valeur numérique	SO	Spécifie la taille logique des volumes. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété <code>volsize</code>” à la page 154.
<code>volblocksize</code>	Valeur numérique	8 Kbytes	(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 Kilo-octets est correcte. Cette propriété peut également s'afficher sous la forme du nom de colonne contracté <code>volblock</code> .
<code>zoned</code>	Booléen	SO	Indique si le jeu de données a été ajouté à une zone non 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 tout système de fichiers ajouté. 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 238.
<code>xattr</code>	Booléen	on	Indique si les attributs étendus sont activés ou désactivés pour le système de fichiers. La valeur par défaut est on.

Propriétés ZFS natives en lecture seule

Les propriétés natives en lecture seule peuvent être récupérées, mais il est impossible de les modifier. 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 correspondant est mentionné dans la description figurant dans le [Tableau 6-1](#).

Les propriétés natives en lecture seule sont répertoriées dans cette section et décrites dans le [Tableau 6-1](#).

- `available`
- `creation`
- `mounted`
- `origin`
- `compressratio`

- `referenced`
- `type`
- `used`

Pour plus d'informations sur cette propriété, reportez-vous à la section [“Propriété `used`”](#) à la page 151.

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 ZFS”](#) à la page 50.

Propriété `used`

Quantité d'espace 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. L'espace 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 utilisée sur le parent par un jeu de données, ainsi que la quantité d'espace 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 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 précédemment partagé devient dédié à l'instantané. Il est alors comptabilisé dans l'espace utilisé par cet instantané. L'espace utilisé par un instantané représente ses données uniques. La suppression d'instantanés peut également augmenter l'espace 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 50.

La quantité d'espace utilisée, disponible ou indiquée ne prend pas en compte les modifications en cours d'exécution. Ces modifications sont généralement prises en compte au bout de quelques secondes. La modification d'un disque utilisant `fsync(3c)` ou `O_SYNC` ne garantit pas la mise à jour immédiate des informations concernant l'utilisation de l'espace.

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 158 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 140. À 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 [“Quotas et réservations ZFS”](#) à la page 170.

Certaines propriétés natives définies sont spécifiques à un type de jeu de données. Dans ce cas, le jeu de données correspondant est mentionné dans la description figurant dans le [Tableau 6-1](#). 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 6-1](#).

- `aclinherit`
Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “[Modes de propriétés d'ACL](#)” à la page 197.
- `aclmode`
Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “[Modes de propriétés d'ACL](#)” à la page 197.
- `atime`
- `canmount`
- Somme de contrôle
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `quota`
- `readonly`
- `recordsize`
Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “[Propriété recordsize](#)” à la page 153.
- `refquota`
- `reservation`
- `reservation`
- `sharenfs`
- `setuid`
- `snapdir`
- `volsize`
Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “[Propriété volsize](#)” à la page 154.
- `volblocksize`

- zoned

Propriété canmount

Si cette propriété 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`. Cette propriété est comparable à la définition de la propriété `mountpoint` sur la valeur `none`. Toutefois, le jeu de données possède toujours une propriété `mountpoint` standard susceptible d'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 à proprement parler n'est jamais monté, ni accessible par les utilisateurs. Dans ce cas, le système de fichiers parent sur lequel cette propriété est désactivée sert de *conteneur* afin de pouvoir définir des attributs 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. 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 list -r userpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
userpool	140K	8.24G	24.5K	/export/home
userpool/user1	24.5K	8.24G	24.5K	/export/home/user1
userpool/user2	24.5K	8.24G	24.5K	/export/home/user2

Définir la propriété `canmount` sur `noauto` indique que le jeu de données ne peut être monté que de manière explicite, et non pas de manière automatique. Ce paramètre est utilisé par le logiciel de mise à niveau de Solaris afin que seuls les jeux de données appartenant à l'environnement d'initialisation actif soient montés lors de l'initialisation.

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 de la propriété `recordsize` sur une valeur 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 deux supérieure ou égale à 512 octets et inférieure ou égale à 128 Ko. La modification de la valeur de `recordsize` affecte uniquement les fichiers créés en aval. Cette modification n'affecte pas les fichiers existants.

Cette propriété peut également s'afficher sous la forme du nom de colonne contracté `recsize`.

Propriété `volsize`

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 corruptions de données. Ces symptômes peuvent également survenir lors de la modification et notamment de la réduction de la taille d'un 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'attribut `-s` dans la commande `zfs create -V` ou modifiez la réservation, une fois le volume créé. Le terme *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 235.

Propriétés ZFS définies par l'utilisateur

Outre les propriétés natives standard, 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 :

- contenir le caractère `:` (deux points) afin de les distinguer des propriétés natives ;
- contenir des lettres en minuscules, des chiffres, ainsi que les signes de ponctuation suivants : `',' '+' ',' '_'` ;
- posséder un nom dont la longueur maximale 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.sun." sont réservés à l'usage de Sun Microsystems.

Les valeurs des propriétés définies par l'utilisateur possèdent les caractéristiques suivantes :

- Il s'agit de chaînes arbitraires systématiquement héritées et jamais validées.
- Leur valeur se constitue de 1 024 caractères maximum.

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.

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`.

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 propriétés `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 143](#).

Exemple :

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
pool                                476K  16.5G   21K    /pool
pool/clone                          18K   16.5G   18K    /pool/clone
pool/home                          296K  16.5G   19K    /pool/home
pool/home/marks                    277K  16.5G  277K    /pool/home/marks
pool/home/marks@snap                0     -     277K   -
pool/test                          18K   16.5G   18K    /test
```

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.

Exemple :

```
# zfs list -r pool/home/marks
NAME                                USED  AVAIL  REFER  MOUNTPOINT
pool/home/marks                    277K  16.5G  277K    /pool/home/marks
pool/home/marks@snap                0     -     277K   -
```

La commande `zfs list` s'utilise avec des noms de chemin absolus pour les jeux de données, les instantanés et les volumes. Exemple :

```
# zfs list /pool/home/marks
NAME                                USED  AVAIL  REFER  MOUNTPOINT
pool/home/marks                    277K  16.5G  277K    /pool/home/marks
```

L'exemple suivant illustre la manière d'afficher `tank/home/chua` et tous ses jeux de données descendants.

```
# zfs list -r tank/home/cha
NAME                                USED  AVAIL  REFER  MOUNTPOINT
tank/home/cha                       26.0K  4.81G  10.0K  /tank/home/cha
tank/home/cha/projects              16K   4.81G   9.0K   /tank/home/cha/projects
tank/home/cha/projects/fs1          8K   4.81G   8K     /tank/home/cha/projects/fs1
tank/home/cha/projects/fs2          8K   4.81G   8K     /tank/home/cha/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`, `-f` 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. Les valeurs valides sont les propriétés de jeu de données. 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 143](#). 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.

Les exemples suivants illustrent l'utilisation de la commande `zfs list` permettant d'afficher le nom de jeu de données et des propriétés `sharenfs` et `mountpoint`.

```
# zfs list -o name,sharenfs,mountpoint
NAME                                SHARENFS  MOUNTPOINT
tank                                off       /tank
tank/home                          on        /tank/home
tank/home/ahrens                   on        /tank/home/ahrens
tank/home/bonwick                  on        /tank/home/bonwick
tank/home/cha                      on        /tank/home/cha
tank/home/eschrock                 on        legacy
tank/home/moore                    on        /tank/home/moore
tank/home/tabriz                   ro        /tank/home/tabriz
```

L'option `-t` permet de spécifier le type de jeu de données à afficher. Les types corrects sont décrits dans le tableau suivant.

TABLEAU 6-2 Types de jeux de données ZFS

Type	Description
filesystem	Systèmes de fichiers et clones
Volume	Volumes

TABLEAU 6-2 Types de jeux de données ZFS (Suite)

Type	Description
Instantané	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 -t filesystem -o name,used
NAME                USED
pool                476K
pool/clone          18K
pool/home           296K
pool/home/marks    277K
pool/test           18K
```

L'option `-H` permet d'exclure l'en-tête de la commande `zfs list` lors de la génération de la sortie. Lorsque vous spécifiez l'option `-H`, les espaces sont générés sous la forme de tabulations. 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 -H -o name
pool
pool/clone
pool/home
pool/home/marks
pool/home/marks@snap
pool/test
```

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 158
- “Héritage des propriétés ZFS” à la page 159
- “Envoi de requêtes sur les propriétés ZFS” à la page 160

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 151. La commande `zfs set` utilise une séquence de propriété/valeur au format *propriété=valeur* et un nom de jeu de données.

L'exemple suivant illustre la définition de la propriété `atime` sur la valeur `off` pour `tank/home`. Chaque exécution de la commande `zfs set` autorise la définition ou la modification d'une seule propriété.

```
# zfs set atime=off tank/home
```

Vous pouvez également définir les propriétés des systèmes de fichiers une fois ces derniers créés. Exemple :

```
# zfs create -o atime=off tank/home
```

Les suffixes explicites suivants (répertoriés par ordre croissant) permettent de spécifier des propriétés numériques : `BKMGTEZ`. 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 formulations de la commande `zfs set` suivantes correspondent à des expressions numériques équivalentes indiquant que la propriété `quota` doit être définie sur 50 Go sur le système de fichiers `tank/home/marks` :

```
# zfs set quota=50G tank/home/marks
# zfs set quota=50g tank/home/marks
# zfs set quota=50GB tank/home/marks
# zfs set quota=50gb tank/home/marks
```

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

Toutes les propriétés définies, à l'exception des propriétés de quotas et de réservations, héritent de la valeur de leur parent (sauf si un quota ou une réservation est explicitement défini pour le système enfant). 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 parent.

L'exemple suivant illustre l'activation de la compression pour le système de fichiers `tank/home/bonwick` à l'aide de la commande `zfs set`. 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/bonwick
# zfs get -r compression tank
NAME                PROPERTY    VALUE          SOURCE
tank                compression off            default
tank/home           compression off            default
tank/home/bonwick  compression on           local
# zfs inherit compression tank/home/bonwick
# zfs get -r compression tank
NAME                PROPERTY    VALUE          SOURCE
tank                compression off            default
tank/home           compression off            default
tank/home/bonwick  compression off            default
```

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` entraîne la suppression de la valeur de propriété actuelle pour l'ensemble des jeux de données descendants.

Pour plus d'informations sur la commande `zfs`, 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 156. 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 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'emplacement à partir duquel la valeur de propriété a été définie. Le tableau suivant indique la signification des valeurs possibles de la colonne SOURCE.

TABLEAU 6-3 Valeurs possibles de la colonne SOURCE (zfs get)

Valeur	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 <i>nom-jeu-données</i>	La valeur de propriété est héritée du parent spécifié par la chaîne <i>nom-jeu-données</i> .
local	La valeur de propriété a été définie de manière explicite pour ce jeu de données à l'aide de la commande <code>zfs set</code> .
temporary	La valeur de propriété a été définie à l'aide la commande <code>zfs mount</code> spécifiée avec l'option <code>-o</code> et n'est valide que pour la durée de vie 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 167.
-(none)	Cette propriété est en lecture seule. Sa valeur est générée par ZFS.

Le mot-clé `all` permet de récupérer toutes les propriétés de jeu de données. Les exemples suivants illustrent la récupération de toutes les propriétés de jeu de données à l'aide du mot-clé `all` :

```
# zfs get all mpool
NAME  PROPERTY      VALUE          SOURCE
mpool type        filesystem      -
mpool creation   Fri Aug 22 14:24 2008 -
mpool used       6.41G          -
mpool available  26.8G          -
mpool referenced 20.5K          -
mpool compressratio 1.00x         -
mpool mounted    yes            -
mpool quota      none           default
mpool reservation none           default
mpool recordsize 128K           default
mpool mountpoint /mpool         default
mpool sharenfs   off            default
mpool checksum   on             default
mpool compression off            default
```

mpool	atime	on	default
mpool	devices	on	default
mpool	exec	on	default
mpool	setuid	on	default
mpool	readonly	off	default
mpool	zoned	off	default
mpool	snappdir	hidden	default
mpool	aclmode	groupmask	default
mpool	aclinherit	restricted	default
mpool	canmount	on	default
mpool	shareiscsi	off	default
mpool	xattr	on	default
mpool	copies	1	default
mpool	version	3	-
mpool	utf8only	off	-
mpool	normalization	none	-
mpool	casesensitivity	sensitive	-
mpool	vscan	off	default
mpool	nbmand	off	default
mpool	sharesmb	off	default
mpool	refquota	none	default
mpool	refreservation	none	default

Remarque – Les propriétés `casesensitivity`, `nbmand`, `normalization`, `sharemgrutf8only` et `vscan` sont définies sur une valeur fixe et ne sont pas prises en charge dans la version Solaris 10.

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 toutes les propriétés définies localement sur `pool`.

```
# zfs get -s local all pool
NAME          PROPERTY    VALUE      SOURCE
pool         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 jeu de données. Dans l'exemple suivant, les propriétés temporaires de tous les jeux de données définis sur `tank` sont affichées de manière récursive :

```
# zfs get -r -s temporary all tank
NAME          PROPERTY    VALUE      SOURCE
tank/home     atime       off        temporary
tank/home/bonwick atime      off        temporary
tank/home/marks atime       off        temporary
```

L'une des fonctionnalités récemment ajoutées permet d'envoyer des requêtes à l'aide de la commande `zfs get` sans spécifier le système de fichiers cible (la requête porte sur tous les pools ou sur tous les systèmes de fichiers). Exemple :

```
# zfs get -s local all
tank/home           atime           off            local
tank/home/bonwick  atime           off            local
tank/home/marks     quota          50G           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. L'option `-H` indique que toute information d'en-tête doit être omise et que tous les espaces doivent être remplacés par des caractères de tabulation. L'uniformisation des espaces permet de faciliter l'analyse des données. L'option `-o` permet de personnaliser la sortie et d'indiquer la liste des valeurs à inclure dans la sortie en les séparant par une virgule. Toutes les propriétés définies à la section "[Présentation des propriétés ZFS](#)" à la page 143, ainsi que les termes littéraux `name`, `value`, `property` et `source` peuvent être fournis dans la liste `-o`.

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 ci-dessus. Dans l'exemple suivant, les options `-r`, `-o` et `-H` sont spécifiées afin de récupérer le nom du jeu de données ainsi que 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
export/home      5.57G
export/home/marks 1.43G
export/home/maybee 2.15G
```

Montage et partage des systèmes de fichiers ZFS

Cette section décrit le processus de gestion des points de montage et des systèmes de fichiers partagés dans ZFS.

- “Gestion des points de montage ZFS” à la page 164
- “Montage de système de fichiers ZFS” à la page 166
- “Utilisation de propriétés de montage temporaires” à la page 167
- “Démontage des systèmes de fichiers ZFS” à la page 168
- “Activation et annulation du partage des systèmes de fichiers ZFS” à la page 169

Gestion des points de montage ZFS

Par défaut, tous les systèmes de fichiers ZFS sont montés lors de l'initialisation à l'aide du service SMF (Service Management Facility) `svc://system/filesystem/local`. Les systèmes de fichiers sont montés sous */chemin*, où *chemin* correspond au nom du système de fichiers.

Vous pouvez remplacer le point de montage par défaut en définissant la propriété `mountpoint` sur un chemin spécifique à l'aide de la commande `zfs set`. La création du point de montage (si nécessaire) et le montage du système de fichiers sont gérés automatiquement par le système ZFS lors de l'exécution de la commande `zfs mount -a`. Il n'est pas nécessaire de modifier manuellement le fichier `/etc/vfstab`.

La propriété `mountpoint` est héritée. Par exemple, si la propriété `mountpoint` de `pool/home` est définie sur `/export/stuff`, `pool/home/user` hérite de la valeur `/export/stuff/user` pour sa propriété `mountpoint`.

La propriété `mountpoint` peut être définie sur `none` afin d'empêcher le montage du système de fichiers. La propriété `canmount` permet également de déterminer si les systèmes de fichiers peuvent être montés. Pour plus d'informations sur la propriété `canmount`, reportez-vous à la section “Propriété `canmount`” à la page 153.

Si vous le souhaitez, vous avez également la possibilité de gérer les systèmes de fichiers de manière explicite à l'aide d'interfaces de montage héritées en définissant la propriété `mountpoint` sur `legacy` dans la commande `zfs set`. Dans ce cas, le montage et la gestion du 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 165.

Lorsque vous modifiez les stratégies de gestion des points de montage, les comportements suivants s'appliquent :

- comportement de point de montage automatique ;
- comportement de point de montage hérité.

Points de montage automatiques

- Si la propriété `legacy` ou `none` est remplacée par une autre valeur, 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.

Vous pouvez également définir le point de montage par défaut du jeu de données racine 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 d'un pool de stockage ZFS” à la page 97](#).

Les jeux de données dont la propriété `mountpoint` n'est pas définie sur `legacy` sont gérés par le système ZFS. L'exemple suivant illustre la création d'un jeu de données dont le point de montage est géré automatiquement 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. Si nécessaire, les répertoires de point de montage sont créés automatiquement. 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`. Les systèmes de fichiers ZFS hérités ne sont pas montés automatiquement lors de l'initialisation et les

commandes ZFS `mount` et `umount` ne s'appliquent pas à ces jeux de données. Les exemples suivants illustrent les commandes de définition et de gestion d'un jeu de données ZFS hérité :

```
# zfs set mountpoint=legacy tank/home/eschrock
# mount -F zfs tank/home/eschrock /mnt
```

Vous devez en outre les monter en créant des entrées dans le fichier `/etc/vfstab`. Sinon, le service `system/filesystem/local` entre en mode de maintenance à l'initialisation du système.

Pour monter automatiquement un système de fichiers hérité, vous devez ajouter une entrée au fichier `/etc/vfstab`. L'exemple suivant illustre une forme possible de l'entrée ajoutée au fichier `/etc/vfstab` :

```
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot  options
#
tank/home/eschrock -      /mnt      zfs      -      yes      -
```

Les entrées `device to fsck` et `fsck pass` sont définies sur `-`. Cette syntaxe s'explique par l'absence d'application de la commande `fsck` aux systèmes de fichiers ZFS. Pour plus d'informations sur l'intégrité des données et sur l'absence d'application de la commande `fsck` au système ZFS, reportez-vous à la section [“Sémantique transactionnelle” à la page 35](#).

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. La commande `zfs mount` s'utilise uniquement lors de la modification des options de montage ou lors du montage ou du démontage explicite de 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. Exemple :

```
# zfs mount
tank                /tank
tank/home           /tank/home
tank/home/bonwick   /tank/home/bonwick
tank/ws             /tank/ws
```

L'option `-a` permet de monter tous les systèmes de fichiers ZFS. Les systèmes de fichiers hérités ne sont pas montés. Exemple :

```
# zfs mount -a
```

Par défaut, le système ZFS autorise uniquement le montage sur les répertoires vides. Pour forcer une opération de montage effectuée sur un répertoire non vide, spécifiez l'option `-O`. Exemple :

```
# zfs mount tank/home/lalt
cannot mount '/export/home/lalt': directory is not empty
use legacy mountpoint to allow this behavior, or use the -0 flag
# zfs mount -0 tank/home/lalt
```

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. Exemple :

```
# zfs mount pool/home/billm
cannot mount 'pool/home/billm': legacy mountpoint
use mount(1M) to mount this filesystem
# mount -F zfs tank/home/billm
```

Le montage d'un système de fichiers requiert l'utilisation d'un ensemble d'options basées sur les valeurs des propriétés associées au jeu de données. Le tableau ci-dessous illustre la corrélation entre les propriétés et les options de montage :

Propriété	Options de montage
devices	devices/nodevices
exec	exec/noexec
readonly	ro/rw
setuid	setuid/nosetuid

L'option de montage nosuid représente un alias de `nodevices`, `nosetuid`.

Utilisation de propriétés de montage temporaires

Si les options 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 jeu de données 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/perrin` :

```
# zfs mount -o ro tank/home/perrin
```

Dans cet exemple, le système de fichiers est censé être démonté. Pour modifier temporairement une propriété d'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 tank/home/perrin
# zfs get atime tank/home/perrin
NAME          PROPERTY      VALUE          SOURCE
tank/home/perrin atime         off            temporary
```

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 peut s'effectuer à l'aide de la commande `zfs unmount`. La commande `unmount` peut utiliser le point de montage ou le 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 tank/home/tabriz
```

L'exemple suivant illustre le démontage d'un système de fichiers avec l'argument de point de montage :

```
# zfs unmount /export/home/tabriz
```

Si le système de fichiers est actif ou 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/eschrock
cannot unmount '/export/home/eschrock': Device busy
# zfs unmount -f tank/home/eschrock
```

Pour garantir la compatibilité ascendante, vous pouvez démonter les systèmes de fichiers ZFS à l'aide de la commande héritée `umount`. Exemple :

```
# umount /export/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

Comme pour les points de montage, ZFS permet de partager automatiquement les systèmes de fichiers. Pour cela, vous devez spécifier la propriété `sharenfs`. Cette méthode 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 droits `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, le partage est annulé pour tous les systèmes de fichiers. 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/eschrock
```

La propriété 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/bricker
# zfs create tank/home/tabriz
# zfs set sharenfs=ro tank/home/tabriz
```

`tank/home/bricker` et `tank/home/tabriz` sont partagés en écriture dès leur création, car ils héritent de la propriété `sharenfs` définie pour `tank/home`. Une fois la propriété définie sur `ro` (lecture seule), `tank/home/tabriz` est partagé en lecture seule, quelle que soit la propriété `sharenfs` définie pour `tank/home`.

Annulation du partage des systèmes de fichiers ZFS

L'activation et l'annulation du partage de la plupart de systèmes de fichiers s'effectuent lors de l'initialisation, de la création ou de la destruction. Toutefois, il peut parfois s'avérer nécessaire d'annuler explicitement le partage des systèmes de fichiers. Ce type d'opération s'effectue à l'aide de la commande `zfs unshare`. Exemple :

```
# zfs unshare tank/home/tabriz
```

Cette commande entraîne l'annulation du partage du système de fichiers `tank/home/tabriz`. 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

Le comportement automatique du système ZFS (partage à l'initialisation et à la création) convient à la plupart des opérations classiques. 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`. Exemple :

```
# zfs share tank/home/tabriz
```

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 classique `mount`, les commandes `share` et `unshare` peuvent être exécutées sur les systèmes de fichiers ZFS. Vous pouvez dès lors partager manuellement les systèmes 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. Dans certains cas cependant, il peut s'avérer préférable de contrôler le partage des systèmes de fichiers à l'aide du modèle classique.

Quotas et réservations ZFS

ZFS prend en charge les quotas et les réservations au niveau des systèmes de fichiers. La propriété `quota` permet de limiter la quantité d'espace disponible pour un système de fichiers. La propriété `reservation` permet quant à elle de garantir la disponibilité d'une certaine quantité d'espace pour un système de fichiers. Ces deux propriétés s'appliquent au jeu de données sur lequel elles sont définies ainsi qu'à ses descendants.

Par exemple, si un quota est défini pour le jeu de données `tank/home`, la quantité d'espace totale utilisée par `tank/home` *et par tous 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 utilisée par un jeu de données et par tous ses descendants est indiquée par la propriété `used`.

Outre les propriétés `quota` et `reservation`, les propriétés `refquota` et `refreservation` vous permettent de gérer l'espace d'un système de fichiers sans prendre en compte l'espace utilisé par les descendants, notamment les instantanés et les clones.

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 utilisé par les jeux de données.
- Les propriétés `refquota` et `refreservation` conviennent à la gestion de l'espace utilisé par les jeux de données et les instantanés.
- La définition d'une valeur `refquota` ou `refreservation` supérieure à une valeur "reservation" n'a aucun effet. Lorsque vous définissez la propriété `quota` ou la propriété `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`. Si certains blocs d'instantanés sont modifiés, la valeur `quota` risque d'être dépassée avant la valeur `refquota`.

Pour plus d'informations, reportez-vous aux exemples ci-dessous.

Définitions de quotas sur les systèmes de fichiers ZFS

La définition et l'affichage des quotas ZFS s'effectuent respectivement à l'aide des commandes `zfs set` et `zfs get`. Dans l'exemple suivant, un quota de 10 Go est défini pour `tank/home/bonwick`.

```
# zfs set quota=10G tank/home/bonwick
# zfs get quota tank/home/bonwick
NAME                PROPERTY      VALUE          SOURCE
tank/home/bonwick  quota        10.0G         local
```

Les quotas ZFS affectent également la sortie des commandes `zfs list` et `df`. Exemple :

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home           16.5K 33.5G  8.50K  /export/home
tank/home/bonwick  15.0K 10.0G  8.50K  /export/home/bonwick
tank/home/bonwick/ws 6.50K 10.0G  8.50K  /export/home/bonwick/ws
# df -h /export/home/bonwick
```

Filesystem	size	used	avail	capacity	Mounted on
tank/home/bonwick	10G	8K	10G	1%	/export/home/bonwick

tank/home dispose de 33,5 Go d'espace. Toutefois, tank/home/bonwick et tank/home/bonwick/ws disposent uniquement de 10 Go d'espace, en raison du quota défini pour tank/home/bonwick.

Vous ne pouvez pas définir un quota sur une valeur inférieure à la quantité d'espace actuellement utilisée par un jeu de données. Exemple :

```
# zfs set quota=10K tank/home/bonwick
cannot set quota for 'tank/home/bonwick': size is less than current used or reserved space
```

Vous pouvez définir une valeur refquota pour un jeu de données qui limite la quantité d'espace qu'il peut utiliser. Cette limite maximale n'inclut pas l'espace utilisé par les instantanés et les clones. Exemple :

```
# zfs set refquota=10g students/studentA
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
profs                106K  33.2G  18K    /profs
students             57.7M 33.2G  19K    /students
students/studentA   57.5M 9.94G  57.5M  /students/studentA
# zfs snapshot students/studentA@today
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
profs                106K  33.2G  18K    /profs
students             57.7M 33.2G  19K    /students
students/studentA   57.5M 9.94G  57.5M  /students/studentA
students/studentA@today  0     -     57.5M  -
```

Par souci de commodité, vous pouvez définir un autre quota pour un jeu de données afin de vous aider à gérer l'espace utilisé par les instantanés. Exemple :

```
# zfs set quota=20g students/studentA
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
profs                106K  33.2G  18K    /profs
students             57.7M 33.2G  19K    /students
students/studentA   57.5M 9.94G  57.5M  /students/studentA
students/studentA@today  0     -     57.5M  -
```

Dans ce scénario, studentA peut atteindre la limite maximale de refquota (10 Go) mais peut supprimer des fichiers pour libérer de l'espace même en présence d'instantanés.

Dans l'exemple ci-dessus, le plus petit des deux quotas (10 Go par opposition à 20 Go) s'affiche dans la sortie de `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. Exemple :

```
# zfs get refquota,quota students/studentA
NAME                PROPERTY  VALUE          SOURCE
students/studentA  refquota  10G            local
students/studentA  quota     20G            local
```

Définition de réservations sur les systèmes de fichiers ZFS

Une *réserve* ZFS désigne une quantité d'espace du pool garantie pour un jeu de données. Dès lors, pour réserver une quantité d'espace 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 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`.

Exemple :

```
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME                PROPERTY  VALUE          SOURCE
tank/home/moore     reservation  5.00G         local
```

Les réservations ZFS peuvent affecter la sortie de la commande `zfs list`. Exemple :

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home           5.00G  33.5G  8.50K  /export/home
tank/home/moore     15.0K  10.0G  8.50K  /export/home/moore
```

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 correspond à l'espace réservé pour `tank/home/moore`. Les réservations sont prises en compte dans l'espace utilisé des jeux de données parent et non dans le quota, la réserve 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 une quantité d'espace supérieure à celle indiquée par sa réserve. Pour cela, le pool doit contenir une quantité d'espace non réservée suffisante et la

quantité d'espace utilisée par le jeu de données doit être inférieur au quota défini pour ce jeu. Un jeu de données ne peut pas utiliser l'espace réservé pour 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.

```
# zfs set reservation=10G tank/home/moore
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME                PROPERTY          VALUE              SOURCE
tank/home/moore     reservation       5.00G             local
```

Vous pouvez définir une valeur `refreservation` afin de garantir l'espace pour un jeu de données qui n'inclut pas l'espace utilisé par les instantanés ni les clones. La valeur `refreservation` est prise en compte dans l'espace utilisé des jeux de données parent et vient en déduction des quotas et réservations des jeux de données parent. 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. 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 l'espace utilisé du parent.

Dans l'exemple ci-dessus, le plus petit des deux quotas (10 Go par opposition à 20 Go) s'affiche dans la sortie de `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. 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 est disponible dans le pool au-delà de cette réservation afin de pouvoir contenir le nombre actuel d'octets *référéncés* dans le jeu de données.

Utilisation des instantanés et des clones ZFS

Ce chapitre fournit des informations sur la création et la gestion d'instantanés et de clones ZFS. Il contient également des informations relatives à l'enregistrement d'instantanés.

Il contient les sections suivantes :

- “Présentation des instantanés ZFS” à la page 177
- “Création et destruction d'instantanés ZFS” à la page 178
- “Affichage et accès des instantanés ZFS” à la page 180
- “Restauration d'un instantané ZFS” à la page 181
- “Présentation des clones ZFS” à la page 182
- “Création d'un clone ZFS” à la page 182
- “Destruction d'un clone ZFS” à la page 183
- “Envoi et réception de données ZFS” à la page 184

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 de libérer de l'espace.

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.
- Aucune sauvegarde de secours distincte n'est utilisée. Les instantanés consomment de l'espace disque provenant directement du pool de stockage auquel appartient le système de fichiers 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 de sélection instantanée, les opérations atomiques assurent ainsi la cohérence des données, 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 184.

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
```

Le nom de l'instantané doit respecter les conventions d'attribution de noms décrites dans la section [“Exigences d'attribution de noms de composants ZFS”](#) à la page 40.

Dans l'exemple suivant, un instantané de `tank/home/ahrens` nommé `friday` est créé.

```
# zfs snapshot tank/home/ahrens@friday
```

Vous pouvez créer des instantanés pour tous les systèmes de fichiers descendants à l'aide de l'option `-r`. Exemple :

```
# zfs snapshot -r tank/home@now
# zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home@now        0      - 29.5K  -
tank/home/ahrens@now 0      - 2.15M  -
tank/home/anne@now   0      - 1.89M  -
tank/home/bob@now    0      - 1.89M  -
tank/home/cindys@now 0      - 2.15M  -
```

Les propriétés des instantanés ne sont pas modifiables. Les propriétés des jeux de données ne peuvent pas être appliquées à un instantané.

```
# zfs set compression=on tank/home/ahrens@tuesday
cannot set compression property for 'tank/home/ahrens@tuesday': snapshot
properties cannot be modified
```

La commande `zfs destroy` permet de détruire les instantanés. Exemple :

```
# zfs destroy tank/home/ahrens@friday
```

La destruction d'un jeu de données est impossible s'il existe des instantanés du jeu de données.
Exemple :

```
# zfs destroy tank/home/ahrens
cannot destroy 'tank/home/ahrens': filesystem has children
use '-r' to destroy the following datasets:
tank/home/ahrens@tuesday
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
```

En outre, afin de pouvoir détruire l'instantané utilisé pour créer les clones, vous devez au préalable détruire les clones.

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 141.

Renommage d'instantanés ZFS

Vous pouvez renommer les instantanés, mais vous devez procéder au sein du pool et du jeu de données à partir desquels ils ont été créés. Exemple :

```
# zfs rename tank/home/cindys@083006 tank/home/cindys@today
```

En outre, le raccourci de syntaxe suivant donne une syntaxe de renommage d'instantané similaire à l'exemple ci-dessus.

```
# zfs rename tank/home/cindys@083006 today
```

L'opération de renommage 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/cindys@today pool/home/cindys@Saturday
cannot rename to 'pool/home/cindys@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`. Exemple :

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPPOINT
users	270K	16.5G	22K	/users
users/home	76K	16.5G	22K	/users/home
users/home@yesterday	0	-	22K	-
users/home/markm	18K	16.5G	18K	/users/home/markm
users/home/markm@yesterday	0	-	18K	-
users/home/marks	18K	16.5G	18K	/users/home/marks

```

users/home/marks@yesterday      0      -      18K  -
users/home/neil                  18K    16.5G   18K  /users/home/neil
users/home/neil@yesterday       0      -      18K  -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -r users/home
NAME                               USED    AVAIL    REFER  MOUNTPOINT
users/home                         76K    16.5G   22K    /users/home
users/home@2daysago               0      -      22K    -
users/home/markm                  18K    16.5G   18K    /users/home/markm
users/home/markm@2daysago        0      -      18K    -
users/home/marks                  18K    16.5G   18K    /users/home/marks
users/home/marks@2daysago        0      -      18K    -
users/home/neil                   18K    16.5G   18K    /users/home/neil
users/home/neil@2daysago         0      -      18K    -

```

Affichage et accès des instantanés ZFS

Les instantanés des systèmes de fichiers sont accessibles dans le répertoire `.zfs/snapshot` au sein de la racine du système de fichiers qui le contient. Par exemple, si `tank/home/ahrens` est monté sur `/home/ahrens`, les données de l'instantané `tank/home/ahrens@thursday` sont accessibles dans le répertoire `/home/ahrens/.zfs/snapshot/thursday`.

```

# ls /tank/home/ahrens/.zfs/snapshot
tuesday wednesday thursday

```

Vous pouvez répertorier les instantanés comme suit :

```

# zfs list -t snapshot
NAME                               USED    AVAIL    REFER  MOUNTPOINT
pool/home/anne@monday              0      -      780K   -
pool/home/bob@monday                0      -      1.01M  -
tank/home/ahrens@tuesday            8.50K   -      780K   -
tank/home/ahrens@wednesday          8.50K   -      1.01M  -
tank/home/ahrens@thursday           0      -      1.77M  -
tank/home/cindys@today              8.50K   -      524K   -

```

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@now                       Wed Aug 27 16:35 2008
tank/home/ahrens@tuesday            Wed Aug 27 16:35 2008
tank/home/ahrens@wednesday          Wed Aug 27 16:35 2008
tank/home/ahrens@thursday           Wed Aug 27 16:36 2008
tank/home/cindys@now                Wed Aug 27 16:37 2008

```

Comptabilisation de l'espace d'instantanés

Lors de la création d'un instantané, son espace 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 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 dédié à d'autres instantanés (et, par conséquent, *utilisé* par ceux-ci).

La propriété `referenced` de l'espace d'un instantané est identique à celle du système de fichiers à la création de l'instantané.

Restauration d'un instantané ZFS

La commande `zfs rollback` permet de supprimer toutes les modifications effectuées 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, le système de fichiers `tank/home/ahrens` revient à l'instantané `tuesday` :

```
# zfs rollback tank/home/ahrens@tuesday
cannot rollback to 'tank/home/ahrens@tuesday': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
# zfs rollback -r tank/home/ahrens@tuesday
```

Dans l'exemple ci-dessus, les instantanés `wednesday` et `thursday` sont supprimés en raison de la restauration de l'instantané `tuesday` précédent.

```
# zfs list -r -t snapshot -o name,creation tank/home/ahrens
NAME                                CREATION
tank/home/ahrens@tuesday            Wed Aug 27 16:35 2008
```

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. Vous pouvez d'autre part créer un instantané d'un clone.

- “Création d'un clone ZFS” à la page 182
- “Destruction d'un clone ZFS” à la page 183
- “Remplacement d'un système de fichiers ZFS par un clone ZFS” à la page 183

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. Même en cas de création d'un clone à un autre emplacement de la hiérarchie, l'instantané d'origine ne peut pas être détruit 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 158.

Dans la mesure où un clone partage initialement son espace disque avec l'instantané d'origine, la propriété `used` est initialement égale à zéro. À mesure que le clone est modifié, il utilise de plus en plus d'espace. 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

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 type du nouveau jeu de données (par exemple, système de fichiers ou volume) est le même que celui de l'instantané à partir duquel a été créé le clone. 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/ahrens/bug123` avec 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/ahrens/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. Exemple :

```
# zfs destroy tank/home/ahrens/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 d'*origine* 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 183](#).

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`.

```
# 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            314K  8.24G  25.5K  /tank/test
tank/test/productA  288K  8.24G  288K   /tank/test/productA
tank/test/productA@today  0      -    288K   -
tank/test/productAbeta  0  8.24G  288K   /tank/test/productAbeta
# zfs promote tank/test/productAbeta
# zfs list -r tank/test
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/test            316K  8.24G  27.5K  /tank/test
tank/test/productA  0  8.24G  288K   /tank/test/productA
tank/test/productAbeta  288K  8.24G  288K   /tank/test/productAbeta
tank/test/productAbeta@today  0      -    288K   -
```

Dans la sortie de `zfs list`, vous pouvez voir que la comptabilisation de l'espace du système de fichiers `productA` d'origine a été remplacée par le système de fichiers `productAbeta`.

Pour terminer le processus de remplacement de clone, renommez les systèmes de fichiers.

Exemple :

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/test            316K  8.24G  27.5K  /tank/test
tank/test/productA  288K  8.24G  288K   /tank/test/productA
tank/test/productA@today    0      -    288K   -
tank/test/productAlegacy    0  8.24G  288K   /tank/test/productAlegacy
```

Vous pouvez également supprimer l'ancien système de fichiers si vous le souhaitez. 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.

- “Envoi d'un instantané ZFS” à la page 185
- “Réception d'un instantané ZFS” à la page 186
- “Réplication distante de données ZFS” à la page 190
- “Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde” à la page 190

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 de système de fichier, le cas échéant. Par exemple, cette solution permet de 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 177.

- 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` lorsque vous souhaitez 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.

Envoi d'un instantané ZFS

Vous pouvez utiliser la commande `zfs send` pour envoyer une copie d'un instantané et recevoir cet instantané dans un autre pool du même système ou dans un autre pool d'un système différent utilisé pour stocker les données de sauvegarde. Pour par exemple envoyer l'instantané à un pool différent du même système, employez une syntaxe du type suivant :

```
# zfs send tank/data@snap1 | zfs recv spool/ds01
```

Si vous envoyez le flux de l'instantané à un système différent, envoyez la sortie de la commande `zfs send` à la commande `ssh`. Exemple :

```
host1# zfs send tank/dana@snap1 | ssh host2 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`. Exemple :

```
host1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh host2 zfs recv newtank/dana
```

Le premier argument correspond à l'instantané le plus ancien (*instantané1*) et le second, à l'instantané le plus récent (*instantané2*). Dans ce cas, le système de fichiers `newtank/dana` doit exister pour que la réception incrémentielle s'effectue correctement.

La source de l'*instantané1* 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 l'*instantané1*, qui est considéré comme provenant du même système de fichiers que l'*instantané2*. Exemple :

```
host1# zfs send -i snap1 tank/dana@snap2 > ssh host2 zfs recv newtank/dana
```

Cette syntaxe est équivalente à l'exemple de syntaxe incrémentielle ci-dessus.

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, vous avez la possibilité de compresser une représentation de flux d'instantané ZFS à l'aide de la commande `gzip`. 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 le nom du système de fichiers choisi existe déjà, la commande `zfs rename` permet de renommer le système de fichiers.

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
```

Vous pouvez utiliser `zfs recv` en tant qu'alias pour la commande `zfs receive`.

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.

Par exemple, si vous effectuez les modifications dans le système de fichiers comme ci-dessous :

```
host2# rm newtank/dana/file.1
```

Si vous effectuez un envoi incrémentiel de tank/dana@snap3, vous devez tout d'abord restaurer le système de fichiers pour recevoir le nouvel instantané incrémentiel. L'option -F permet d'éviter l'étape de restauration. Exemple :

```
host1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh host2 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, le message suivant s'affiche :

```
host1# zfs send -i tank/dana@snap4 tank/dana@snap5 | ssh host2 zfs recv newtank/dana
cannot receive: destination has been modified since most recent snapshot
```

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 à 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
```

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 :

- 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 reçus, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.
- Vous pouvez également utiliser les deux options pour envoyer un flux de réplication incrémentiel.

- Les modifications apportées aux propriétés, ainsi que les changements de nom et destructions d'instantané et de système de fichiers sont conservés.
- Si l'option `zfs recv -F` n'est pas spécifiée lors de la réception du flux de réplication, les destructions de jeu de données sont ignorées. La syntaxe de `zfs recv -F` dans ce cas peut conserver également sa signification de *recupération le cas échéant*.
- Tout comme dans les autres cas (autres que `zfs send -R`) - `i` ou -`I`, si l'option -`I` est utilisée, tous les instantanés créés entre instantanéA et instantanéD sont envoyés. Si l'option -`i` est utilisée, seul l'instantanéD (de 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 7-1 Envoi et réception de flux d'instantanés ZFS complexes (exemples)

Plusieurs instantanés incrémentiels peuvent être regroupés en un seul instantané à l'aide de l'option `zfs send -I`. Exemple :

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

Supprimez les instantanés B, C et D.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Recevez l'instantané regroupé.

```
# 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. Exemple :

```
# zfs create pool/fs
# zfs snapshot pool/fs@snap1
# zfs clone pool/fs@snap1 pool/clone
```

EXEMPLE 7-1 Envoi et réception de flux d'instantanés ZFS complexes (exemples) (Suite)

```
# 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
```

Utilisez 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 reçus, 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
# 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    -
```

Vous pouvez utiliser la commande `zfs send -R` pour répliquer le jeu de données `users` et ses descendants, puis envoyer le flux répliqué à 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
```

EXEMPLE 7-1 Envoi et réception de flux d'instantanés ZFS complexes (exemples) (Suite)

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. Exemple :

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Cette commande envoie les données de l'instantané `tank/cindy@today`, le reçoit dans le système de fichiers `sandbox/restfs` et crée également un instantané `restfs@today` sur le système `newsys`. Dans cet exemple, l'utilisateur a été configuré pour utiliser `ssh` dans le système distant.

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. Tous ces utilitaires enregistrent et restaurent les attributs de fichiers et les ACL ZFS. Vérifiez les options adéquates des commandes `tar` et `cpio`.

Pour connaître les toutes dernières informations relatives à ZFS et aux problèmes concernant les produits de sauvegarde tiers, consultez les notes de version de Solaris 10 ou la FAQ de ZFS à l'adresse :

<http://opensolaris.org/os/community/zfs/faq/#backupsoftware>

Utilisation des ACL pour la protection de fichiers ZFS

Ce chapitre décrit l'utilisation des listes de contrôle d'accès (ACL, Access Control List) pour protéger les fichiers ZFS en accordant plus de droits granulaires que de droits UNIX standard.

Il contient les sections suivantes :

- “Nouveau modèle ACL Solaris” à la page 191
- “Configuration d'ACL dans des fichiers ZFS” à la page 198
- “Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé” à la page 200
- “Configuration et affichage d'ACL dans des fichiers ZFS en format compact” à la page 216

Nouveau modèle ACL Solaris

Les versions précédentes les plus récentes 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 8-2](#).
- 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 196.

Les deux modèles d'ACL assurent un contrôle d'accès bien plus fin, et disponible avec les droits 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 les droits autorisés et celles qui ne le sont pas. 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 de droits si les droits qui n'ont pas été définis dans cette ACE sont ou non autorisés.

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 `ufs restore` 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 NFSv4 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 190.

Descriptions de syntaxe pour la configuration des ACL

Deux formats d'ACL de base sont fournis comme suit :

Syntax for Setting Trivial ACLs

`chmod [options] A[index]{+|=}owner@ |group@ |everyone@: droits d'accès/...[:indicateurs d'héritage]: deny | allow fichier`

`chmod [options] A-owner@, group@, everyone@: droits d'accès /...[:indicateurs d'héritage]:deny | allow fichier ...`

`chmod [options] A[index]- fichier`

Syntaxe pour la configuration d'ACL non triviales

`chmod [options] A[index]{+|=}user|group:name: droits d'accès /...[:indicateurs d'héritage]:deny | allow fichier`

`chmod [options] A-user|group:name: droits d'accès /...[:indicateurs d'héritage]:deny | allow fichier ...`

`chmod [options] A[index]- fichier`

`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 8-1](#).

utilisateur ou groupe :*ID-entrée-ACL=nomutilisateur ou nomgroupe*

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 8-1](#).

droits-accès/.../

Identifie les droits d'accès accordés ou refusés. Pour obtenir une description des privilèges d'accès d'ACL, reportez-vous au [Tableau 8-2](#).

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 8-3](#).

deny | allow

Détermine si les droits d'accès sont accordés ou refusés.

Dans l'exemple suivant, la valeur de l'*ID d'entrée d'ACL* n'est pas pertinente.

```
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 8-1 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' <i>ID d'entrée d'ACL</i> qui contient un <i>nom d'utilisateur</i> ou un <i>ID utilisateur</i> . Le type d'entrée d'ACL est incorrect si la valeur n'est ni un UID numérique, ni un <i>nom d'utilisateur</i> .
group	Avec un nom de groupe, spécifie l'accès accordé à un utilisateur supplémentaire de l'objet. Doit inclure l' <i>ID d'entrée d'ACL</i> qui contient un <i>nom de groupe</i> ou un <i>ID de groupe</i> . Le type d'entrée d'ACL est incorrect si la valeur n'est ni un GID numérique, ni un <i>nom de groupe</i> .

Les privilèges d'accès sont décrits dans le tableau suivant.

TABLEAU 8-2 Privilèges d'accès d'ACL

Privilège d'accès	Privilège d'accès compact	Description
add_file	w	Droit d'ajouter un fichier à un répertoire.
add_subdirectory	p	Dans un répertoire, droit de créer un sous-répertoire.
append_data	p	Marque de réservation. Non implémentée actuellement.
delete	D	Droit de supprimer un fichier.
delete_child	D	Droit de supprimer un fichier ou un répertoire au sein d'un répertoire.
exécution	x	Droit d'exécuter un fichier ou d'effectuer une recherche dans le contenu d'un répertoire.
list_directory	r	Droit de dresser la liste du contenu d'un répertoire.
read_acl	C	Droit de lire l'ACL (1s).
read_attributes	a	Droit 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 <code>ls(1)</code> et <code>stat(2)</code> .
read_data	r	Droit de lire le contenu du fichier.
read_xattr	R	Droit de lire les attributs étendus d'un fichier ou d'effectuer une recherche dans le répertoire d'attributs étendus d'un fichier.
synchroniser	s	Marque de réservation. Non implémentée actuellement.
write_xattr	W	Droit de créer des attributs étendus ou d'écrire dans le répertoire d'attributs étendus. L'attribution de ce droit à un utilisateur signifie que ce dernier peut créer un répertoire d'attributs étendus pour un fichier. Les droits du fichier d'attributs contrôlent l'accès de l'utilisateur à l'attribut.
write_data	w	Droit de modifier ou de remplacer le contenu d'un fichier.
write_attributes	A	Droit de remplacer les durées associées à un fichier ou un répertoire par une valeur arbitraire.
write_acl	C	Droit d'écriture sur l'ACL ou capacité de la modifier à l'aide de la commande <code>chmod</code> .

TABLEAU 8-2 Privilèges d'accès d'ACL (Suite)

Privilège d'accès	Privilège d'accès compact	Description
write_owner	O	Droit de modifier le propriétaire ou le groupe d'un fichier. Ou capacité d'exécuter les commandes <code>chown</code> ou <code>chgrp</code> sur le fichier. Droit de devenir propriétaire d'un fichier ou droit de définir la propriété de groupe du fichier sur un groupe dont fait partie l'utilisateur. Le privilège <code>PRIV_FILE_CHOWN</code> est requis pour définir la propriété de fichier ou de groupe sur un groupe ou un utilisateur arbitraire.

Héritage d'ACL

L'héritage d'ACL a pour finalité de permettre à un fichier ou répertoire récemment créé d'hériter des ACL qui leurs sont destinées, tout en tenant compte des bits de droits 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 subséquent 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 8-3 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 <code>file_inherit</code> et/ou <code>dir_inherit</code> 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 <code>file_inherit</code> et/ou <code>dir_inherit</code> afin de spécifier ce qui doit être hérité.
-	SO	Aucun droit n'est accordé.

De plus, vous pouvez configurer un héritage d'ACL par défaut plus ou moins strict 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.

Modes de propriétés d'ACL

Le système de fichiers ZFS inclut deux modes de propriétés relatifs aux ACL :

- `aclinherit` – Cette propriété détermine le comportement de l'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 de droit du fichier ou répertoire.
 - `noallow` – Pour les nouveaux objets, seules les entrées d'ACL héritables dont le type d'accès est `deny` sont héritées.
 - `secure restricted` – Pour les nouveaux objets, les droits `write_owner` et `write_acl` sont supprimés 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.

Le mode par défaut de `aclinherit` est `securerestricted`.

- `aclmode` – Cette propriété modifie le comportement des ACL lorsqu'un fichier est créé ou chaque fois que le mode d'un fichier ou d'un répertoire est modifié à l'aide de la commande `chmod`. Les valeurs possibles sont les suivantes :
 - `discard` – Toutes les entrées d'ACL sont supprimées à l'exception des entrées nécessaires à la définition du mode pour le fichier ou le répertoire.
 - `groupmask` – Les droits d'ACL de groupe ou d'utilisateur sont réduits de manière à ce qu'ils ne soient pas supérieurs aux bits de droit du groupe, à moins qu'il ne s'agisse d'une entrée d'utilisateur ayant le même UID que le propriétaire du fichier ou du répertoire. Ensuite, les droits d'ACL sont réduits afin qu'ils ne dépassent pas les bits de droit du propriétaire.
 - `passthrough` – Au cours d'une opération `chmod`, les ACE autres que `owner@`, `group@` ou `everyone@` ne sont modifiées d'aucune manière. Les ACE `owner@`, `group@` ou `everyone@` sont désactivées afin de définir le mode de fichier comme demandé par l'opération `chmod`.

Le mode par défaut de la propriété `aclmode` est `groupmask`.

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. L'ACL est habituellement *insignifiante* dans la mesure où elle ne représente que les entrées UNIX traditionnelles owner/group/other.

Les fichiers ZFS disposent toujours de bits de droit et d'un mode, mais ces valeurs constituent plus un cache de ce que représente une ACL. Par conséquent, si vous modifiez les droit du fichier, l'ACL du fichier 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 de droit qui accordent l'accès à un groupe ou à tous. L'ensemble des décisions de contrôle d'accès est régi par les droit représentés 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 le droit allow accordé, ce dernier ne peut plus être refusé par la suite par une entrée d'ACL de refus dans le même jeu de droits d'ACL.
- Le propriétaire du fichier dispose du droit `write_acl` de façon inconditionnelle, même si celui-ci est explicitement refusé. Dans le cas contraire, tout droit non spécifié est refusé.

Dans les cas de droits deny ou lorsqu'un droit d'accès est manquant, le sous-système de privilèges détermine la requête 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, consultez le [Tableau 8-3](#) et la section "Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé" à la page 206.

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
-r--r--r--  1 root   root       206663 May  4 11:52 file.1
  0:owner@:write_data/append_data/execute:deny
  1:owner@:read_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

```

Notez que chaque catégorie d'utilisateur (`owner@`, `group@`, `everyone@`) de cet exemple dispose de deux entrées d'ACL. Une entrée correspond aux droits `deny` et une autre, aux droits `allow`.

Voici une description de l'ACL de ce fichier :

- 0:owner@ Les droits d'écriture et d'exécution sur fichier sont refusés au propriétaire (`write_data/append_data/execute:deny`).
- 1:owner@ Le propriétaire peut lire et modifier le contenu d'un 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`). De plus, le propriétaire peut modifier la propriété du fichier (`write_owner:allow`).
- 2:group@ Les droits de modification et d'exécution sur fichier sont refusés au groupe (`write_data/append_data/execute:deny`).
- 3:group@ Des droits de lecture du fichier sont accordés au groupe (`read_data:allow`).
- 4:everyone@ Le droit d'exécuter ou de modifier le contenu ou tout attribut d'un fichier est refusé à toute personne ne correspondant ni à un groupe ni à un utilisateur (`write_data/append_data/write_xattr/execute/write_attributes/write_acl/write_owner:deny`).
- 5:everyone@ Les droits 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`). Le droit d'accès `synchronize` n'est actuellement pas implémenté.

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 Feb 23 10:37 dir.1
0:owner@:deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr

```

```
/write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Voici une description de l'ACL de ce répertoire :

- 0:owner@ La liste de refus du propriétaire est vide pour le répertoire (: : deny).
- 1:owner@ Le propriétaire peut lire et modifier le contenu du répertoire (list_directory/read_data/add_file/write_data/add_subdirectory/append_data), effectuer des recherches dans le contenu (execute) et modifier les attributs du fichier, notamment les horodatages, les attributs étendus et les ACL (write_xattr/write_attributes/write_acl). De plus, le propriétaire peut modifier la propriété du répertoire (write_owner:allow).
- 2:group@ Le groupe ne peut pas ajouter ni modifier le contenu du répertoire (add_file/write_data/add_subdirectory/append_data :deny).
- 3:group@ Le groupe peut répertorier et lire le contenu du répertoire. De plus, le groupe dispose de droits d'exécution pour effectuer des recherches dans le contenu du répertoire (list_directory/read_data/execute:allow).
- 4:everyone@ Le droit d'ajouter ou de modifier le contenu du répertoire est refusé à toute personne ne correspondant ni à un utilisateur ni à un groupe (add_file/write_data/add_subdirectory/append_data). De plus, le droit de modifier les attributs du répertoire est refusé. (write_xattr/write_attributes/write_acl/write_owner:deny).
- 5:everyone@ Toute personne n'étant ni un utilisateur ni un groupe dispose de droits 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). Le droit d'accès synchronise n'est actuellement pas implémenté.

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 193.

- 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 *ID d'index*

```
% chmod Aindex-ID+acl-specification filename
```

Cette syntaxe insère la nouvelle entrée d'ACL à l'emplacement d'*ID d'index* 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'*ID d'index*

```
% 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`. Exemple :

```
# ls -v file.1
-rw-r--r--  1 root   root      206663 Feb 16 11:00 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
```

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 216.

EXEMPLE 8-1 Modification des ACL triviales dans des fichiers ZFS

Cette section fournit des exemples de définition et d'affichage d'ACL insignifiantes.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.1` :

```
# ls -v file.1
-rw-r--r--  1 root    root      206663 Feb 16 11:00 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
```

Dans l'exemple suivants, les droits `write_data` sont accordés au groupe `group@`.

```
# chmod A2=group@:append_data/execute:deny file.1
# chmod A3=group@:read_data/write_data:allow file.1
# ls -v file.1
-rw-rw-r--  1 root    root      206663 May  3 16:36 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@:append_data/execute:deny
 3:group@:read_data/write_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
   /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
   :allow
```

Dans l'exemple suivant, les droits du fichier `file.1` sont reconfigurés sur `644`.

```
# chmod 644 file.1
# ls -v file.1
-rw-r--r--  1 root    root      206663 May  3 16:36 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
```

EXEMPLE 8-2 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 droits `read_data/execute` sont ajoutés à l'utilisateur `gozer` dans le répertoire `test.dir`.

```
# chmod A+user:gozer:read_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+ 2 root    root          2 Feb 16 11:12 test.dir
 0:user:gozer:list_directory/read_data/execute:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Dans l'exemple suivant, les droits `read_data/execute` sont retirés à l'utilisateur `gozer`.

```
# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x 2 root    root          2 Feb 16 11:12 test.dir
 0:owner@::deny
 1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 2:group@:add_file/write_data/add_subdirectory/append_data:deny
 3:group@:list_directory/read_data/execute:allow
 4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

EXEMPLE 8-3 Interactions entre les ACL et les droits dans les fichiers ZFS

Ces exemples d'ACL illustrent l'interaction entre la configuration d'ACL et la modification des bits de droit du fichier ou du répertoire.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.2`:

EXEMPLE 8-3 Interactions entre les ACL et les droits dans les fichiers ZFS (Suite)

```
# ls -v file.2
-rw-r--r--  1 root    root      2703 Feb 16 11:16 file.2
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
   /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
   /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
   :allow
```

Dans l'exemple suivant, les droits d'ACL (allow) sont retirés à everyone@.

```
# chmod A5- file.2
# ls -v file.2
-rw-r-----  1 root    root      2703 Feb 16 11:16 file.2
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
   /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
   /write_acl/write_owner:deny
```

Dans cette sortie, les bits de droit du fichier sont réinitialisés de 655 à 650. Les droits de lecture de everyone@ ont été supprimés des bits de droit du fichier lorsque les droits "allow" des ACL ont été supprimés de everyone@.

Dans l'exemple suivant, l'ACL existante est remplacée par des droits 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      1532 Feb 16 11:18 file.3
 0:everyone@:read_data/write_data:allow
```

Dans cette sortie, la syntaxe chmod remplace effectivement l'ACL existante par les droits read_data/write_data:allow pour les droits 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 droits pour l'utilisateur ou le groupe, les bits de droit sont définis sur 666.

Dans l'exemple suivant, l'ACL existante est remplacée par des droits de lecture pour l'utilisateur gozer.

EXEMPLE 8-3 Interactions entre les ACL et les droits dans les fichiers ZFS (Suite)

```
# chmod A=user:gozer:read_data:allow file.3
# ls -v file.3
-----+ 1 root    root          1532 Feb 16 11:18 file.3
      0:user:gozer:read_data:allow
```

Dans cette sortie, les droits de fichier sont calculés pour être 000 car aucune entrée d'ACL n'existe pour owner@, group@, ou everyone@, qui représentent les composant de droit classiques d'un fichier. Le propriétaire du fichier peut résoudre ce problème en réinitialisant les droits (et l'ACL) comme suit :

```
# chmod 655 file.3
# ls -v file.3
-rw-r-xr-x+ 1 root    root          0 Mar  8 13:24 file.3
      0:user:gozer::deny
      1:user:gozer:read_data:allow
      2:owner@:execute:deny
      3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
        /write_acl/write_owner:allow
      4:group@:write_data/append_data:deny
      5:group@:read_data/execute:allow
      6:everyone@:write_data/append_data/write_xattr/write_attributes
        /write_acl/write_owner:deny
      7:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
        /synchronize:allow
```

EXEMPLE 8-4 Restauration des ACL triviales dans des fichiers ZFS

Vous pouvez utiliser la commande `chmod` pour supprimer toutes les ACL non insignifiantes d'un fichier ou d'un répertoire.

Dans l'exemple suivant, deux ACE non insignifiantes existent dans `test5.dir`.

```
# ls -dv test5.dir
drwxr-xr-x+ 2 root    root          2 Feb 16 11:23 test5.dir
      0:user:gozer:read_data:file_inherit:deny
      1:user:lp:read_data:file_inherit:deny
      2:owner@::deny
      3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
        /append_data/write_xattr/execute/write_attributes/write_acl
        /write_owner:allow
      4:group@:add_file/write_data/add_subdirectory/append_data:deny
      5:group@:list_directory/read_data/execute:allow
      6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
        /write_attributes/write_acl/write_owner:deny
```

EXEMPLE 8-4 Restauration des ACL triviales dans des fichiers ZFS (Suite)

```
7: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 six valeurs par défaut de owner@, group@ et everyone@.

```
# chmod A- test5.dir
# ls -dv test5.dir
drwxr-xr-x  2 root    root          2 Feb 16 11:23 test5.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

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.

En outre, deux propriétés d'ACL sont fournies afin de permettre leur configuration globale dans les systèmes de fichiers : `aclinherit` et `aclmode`. Par défaut, `aclinherit` est définie sur `secure restricted` et `aclmode`, sur `groupmask`.

Pour plus d'informations, reportez-vous à la section [“Héritage d'ACL” à la page 196](#).

EXEMPLE 8-5 Attribution d'héritage d'ACL par défaut

Par défaut, les ACL ne sont pas propagées par le biais d'une structure de répertoire.

Dans l'exemple suivant, une ACE non insignifiante 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
```

EXEMPLE 8-5 Attribution d'héritage d'ACL par défaut (Suite)

```

0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

Si 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 droits 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 Jun 20 14:37 test.dir/sub.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

EXEMPLE 8-6 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 l'exemple suivant, les droits `read_data/write_data` sont ajoutés pour les fichiers dans le répertoire `test.dir` pour l'utilisateur `gozer` pour qu'il dispose de l'accès 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 Jun 20 14:38 test2.dir
0:user:gozer:read_data/write_data:file_inherit:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory

```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```

    /append_data/write_xattr/execute/write_attributes/write_acl
    /write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

Dans l'exemple suivant, les droits de l'utilisateur gozer sont appliqués 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 Jun 20 14:39 test2.dir/file.2
  0:user:gozer:write_data:deny
  1:user:gozer:read_data/write_data:allow
  2:owner@:execute:deny
  3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:allow
  4:group@:write_data/append_data/execute:deny
  5:group@:read_data:allow
  6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
    /write_acl/write_owner:deny
  7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
    :allow

```

Dans la mesure où la propriété `aclmode` pour ce fichier est paramétrée sur le mode par défaut, `groupmask`, l'utilisateur gozer ne dispose pas du droit `write_data` pour le fichier `file.2` car les droits de groupe du fichier ne le permettent pas.

Notez que le droit `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 le droit des droits `everyone@`, à moins qu'il ne soit le propriétaire du fichier ou membre du groupe propriétaire du fichier. Exemple :

```

# mkdir test2.dir/subdir.2
# ls -dv test2.dir/subdir.2
drwxr-xr-x+ 2 root    root          2 Jun 20 14:40 test2.dir/subdir.2
  0:user:gozer:list_directory/read_data/add_file/write_data:file_inherit
    /inherit_only:allow
  1:owner@::deny

```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```

2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

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.

```

# 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 Jun 20 14:41 test3.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
  :file_inherit/dir_inherit:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

# touch test3.dir/file.3
# ls -v test3.dir/file.3
-rw-r--r--+ 1 root    root          0 Jun 20 14:42 test3.dir/file.3
 0:user:gozer:write_data/execute:deny
 1:user:gozer:read_data/write_data/execute:allow
 2:owner@:execute:deny
 3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
 4:group@:write_data/append_data/execute:deny
 5:group@:read_data:allow
 6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny

```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```

7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow

# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+ 2 root root 2 Jun 20 15:13 test3.dir/subdir.1
0:user:gozer:list_directory/read_data/add_file/write_data/execute
:file_inherit/dir_inherit/inherit_only:allow
1:user:gozer:add_file/write_data:deny
2:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
3:owner@::deny
4:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
5:group@:add_file/write_data/add_subdirectory/append_data:deny
6:group@:list_directory/read_data/execute:allow
7:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny
8:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

Dans ces exemples, les bits de droit du répertoire parent pour `group@` et `everyone@` n'accordent pas les droits. Par conséquent, l'utilisateur `gozer` se voit refuser ces droits. La propriété par défaut `aclmode` est `securerestricted`, ce qui signifie que les droits `write_data` et `execute` ne sont pas hérités.

Dans l'exemple suivant, l'utilisateur `gozer` se voit accorder les droits de lecture, d'écriture et d'exécution qui sont hérités 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-xr-x+ 2 root root 2 Jun 20 14:46 test4.dir
0:user:gozer:list_directory/read_data/add_file/write_data/execute
:file_inherit/no_propagate:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny

```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Comme l'exemple suivant l'illustre, lors de la création d'un sous-répertoire, le droit `read_data/write_data/execute` de l'utilisateur `gozer` pour les fichiers n'est pas propagée au nouveau répertoire `sub4.dir`.

```
mkdir test4.dir/sub4.dir
# ls -dv test4.dir/sub4.dir
drwxr-xr-x  2 root  root          2 Jun 20 15:14 test4.dir/sub4.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Comme l'exemple suivant l'illustre, le droit `read_data/write_data/execute` de `gozer` pour les fichiers est propagé vers le nouveau fichier.

```
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root  root          0 Jun 20 15:22 test4.dir/file.4
0:user:gozer:write_data/execute:deny
1:user:gozer:read_data/write_data/execute:allow
2:owner@:execute:deny
3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
/write_acl/write_owner:allow
4:group@:write_data/append_data/execute:deny
5:group@:read_data:allow
6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
/write_acl/write_owner:deny
7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

EXEMPLE 8-7 Héritage d'ACL avec mode de liste défini sur Passthrough

Si la propriété `aclmode` dans le système de fichier `tank/cindy` est définie sur `passthrough`, l'utilisateur `gozer` hérite alors de l'ACL appliquée à `test4.dir` pour le fichier `file.4` récemment créé, comme suit :

EXEMPLE 8-7 Héritage d'ACL avec mode de liste défini sur Passthrough (Suite)

```
# zfs set aclmode=passthrough tank/cindy
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 Jun 20 15:25 test4.dir/file.4
 0:user:gozer:write_data/execute:deny
 1:user:gozer:read_data/write_data/execute:allow
 2:owner@:execute:deny
 3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
   /write_acl/write_owner:allow
 4:group@:write_data/append_data/execute:deny
 5:group@:read_data:allow
 6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
   /write_acl/write_owner:deny
 7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
   :allow
```

Cette sortie montre que l'ACL

read_data/write_data/execute:allow:file_inherit/dir_inherit définie sur le répertoire parent, test4.dir, est transmise à l'utilisateur gozer.

EXEMPLE 8-8 Héritage d'ACL avec mode de liste défini sur Discard

Si la propriété aclmode d'un système de fichiers est définie sur discard, il est alors possible de supprimer les ACL avec les bits de droit dans un changement de répertoire. Exemple :

```
# zfs set aclmode=discard tank/cindy
# chmod A+user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
# ls -dv test5.dir
drwxr-xr-x+ 2 root    root          2 Feb 16 11:23 test5.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
   :dir_inherit:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
   /append_data/write_xattr/execute/write_attributes/write_acl
   /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
   /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
   /read_acl/synchronize:allow

# zfs set aclmode=discard tank/cindy
# chmod A+user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
```

EXEMPLE 8-8 Héritage d'ACL avec mode de liste défini sur Discard (Suite)

```
# ls -dv test5.dir
drwxr-xr-x+ 2 root    root          2 Jun 20 15:21 test5.dir
0:user:gozer:list_directory/read_data/add_file/write_data/execute
:dir_inherit:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Si vous décidez ultérieurement de renforcer les bits de droit d'un répertoire, l'ACL non triviale est supprimée. Exemple :

```
# chmod 744 test5.dir
# ls -dv test5.dir
drwxr--r-- 2 root    root          2 Jun 20 15:21 test5.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data/execute:deny
3:group@:list_directory/read_data:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/execute/write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow
```

EXEMPLE 8-9 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 autorise le droit `read_data`, tandis qu'un autre refuse ce droit. 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 Jun 20 15:24 test6.dir
0:user:gozer:read_data:file_inherit:deny
```

EXEMPLE 8-9 Héritage d'ACL avec mode d'héritage de liste défini sur Noallow (Suite)

```

1:user:lp:read_data:file_inherit:allow
2:owner@::deny
3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
4:group@:add_file/write_data/add_subdirectory/append_data:deny
5:group@:list_directory/read_data/execute:allow
6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
7:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

Comme l'illustre l'exemple suivant, lors de la création d'un nouveau fichier, l'ACL qui autorise le droit `read_data` est supprimée.

```

# touch test6.dir/file.6
# ls -v test6.dir/file.6
-rw-r--r--  1 root   root           0 Jun 20 15:25 test6.dir/file.6
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow

```

EXEMPLE 8-10 Héritage d'ACL avec mode d'héritage de liste défini sur Passthrough

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 de droit déterminé par les ACE pouvant être héritées. Si aucune ACE pouvant être héritée n'affecte le mode de droit, ce mode est alors défini en fonction du mode demandé à partir de l'application.

Les exemples suivants utilisent la syntaxe ACL compacte pour illustrer le processus d'héritage des bits de droit 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

EXEMPLE 8-10 Héritage d'ACL avec mode d'héritage de liste défini sur Passthrough (Suite)

répertoires nouvellement créés héritent d'une entrée d'ACL @owner, group@ et everyone@. En outre, les répertoires héritent de six autres ACE qui appliquent les ACE aux répertoires et fichiers nouvellement créés.

```
# zfs set aclinherit=passthrough tank/cindys
# pwd
/tank/cindys
# mkdir test1.dir

# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow
test1.dir
# ls -Vd test1.dir
drwxrwx---+ 2 root    root          2 Jul 29 10:56 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.

```
# cd test1.dir
# touch file.1
# ls -V file.1
-rwxrwx---+ 1 root    root          0 Jul 29 10:58 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 Jul 29 10:59 subdir.1
      owner@:rwxpdDaARWcCos:fdi---:allow
      owner@:rwxpdDaARWcCos:-----:allow
      group@:rwxp-----:fdi---:allow
      group@:rwxp-----:-----:allow
      everyone@:-----:fdi---:allow
      everyone@:-----:-----:allow
```

Les entrées -di- et f-i- 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.

EXEMPLE 8-10 Héritage d'ACL avec mode d'héritage de liste défini sur Passthrough (Suite)

```
# cd /tank/cindys
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r--  1 root    root          0 Jul 29 11:15 file.2
  owner@:--x-----:-----:deny
  owner@:rw-p---A-W-Co-:-----:allow
  group@:-wzp-----:-----:deny
  group@:r-----:-----:allow
  everyone@:-wzp---A-W-Co-:-----:deny
  everyone@:r-----a-R-c--s:-----:allow
```

Configuration et affichage d'ACL dans des fichiers ZFS en format compact

Vous pouvez définir et afficher les droits relatifs aux fichiers ZFS en format compact utilisant 14 lettres uniques pour représenter les droits. Les lettres représentant les droits compacts sont répertoriées dans le [Tableau 8-2](#) et le [Tableau 8-3](#).

Vous pouvez afficher les listes d'ACL compactes pour les fichiers et les répertoires à l'aide de la commande `ls -V`. Exemple :

```
# ls -V file.1
-rw-r--r--  1 root    root        206663 Feb 16 11:00 file.1
  owner@:--x-----:-----:deny
  owner@:rw-p---A-W-Co-:-----:allow
  group@:-wzp-----:-----:deny
  group@:r-----:-----:allow
  everyone@:-wzp---A-W-Co-:-----:deny
  everyone@:r-----a-R-c--s:-----:allow
```

La sortie d'ACL compacte est décrite comme suit :

owner@ Le droit d'exécution sur le fichier est refusé au propriétaire (x= execute).

owner@ Le propriétaire peut lire et modifier le contenu du fichier (rw=read_data/write_data), (p= append_data). Il peut également modifier les attributs du fichier, par exemple l'horodatage, les attributs étendus et les ACL (A=write_xattr, W=write_attributes, C= write_acl). De plus, le propriétaire peut modifier la propriété du fichier (o=write_owner).

group@	Les droits de modification et d'exécution sur le fichier sont refusés au groupe (write_data, p=append_data et x=execute).
group@	Les droits de lecture sur le fichier sont accordés au groupe (r= read_data).
everyone@	Les droits d'exécution ou de modification du contenu du fichier, ou de modification de tout attribut du fichier sont refusés à toute personne n'étant ni un utilisateur ni un groupe (w=write_data, x= execute, p=append_data, A=write_xattr, W=write_attributes , C=write_acl et o= write_owner).
everyone@	Les droits 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). Le droit d'accès synchronize n'est actuellement pas implémentée.

Le format d'ACL compact dispose des avantages suivants par rapport au format d'ACL détaillé :

- Les droits peuvent être spécifiés en tant qu'arguments de position pour la commande chmod.
- Les tirets (-), qui n'identifient aucun droit, peuvent être supprimés. Seules les lettres nécessaires doivent être spécifiées.
- Les indicateurs de droits 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 200.

EXEMPLE 8-11 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-xr-x 1 root    root      206663 Feb 16 11:00 file.1
  owner@: --x-----: -----:deny
  owner@: rw-p---A-W-Co-: -----:allow
  group@: -w-p-----: -----:deny
  group@: r-x-----: -----:allow
  everyone@: -w-p---A-W-Co-: -----:deny
  everyone@: r-x---a-R-c--s: -----:allow
```

Dans cet exemple, les droits read_data/execute sont ajoutés pour l'utilisateur gozer dans le fichier `file.1`.

```
# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r-xr-x+ 1 root    root      206663 Feb 16 11:00 file.1
  user:gozer:r-x-----: -----:allow
  owner@: --x-----: -----:deny
  owner@: rw-p---A-W-Co-: -----:allow
```

EXEMPLE 8-11 Configuration et affichage des ACL en format compact (Suite)

```

group@: -w-p-----:-----:deny
group@: r-x-----:-----:allow
everyone@: -w-p---A-W-Co-:-----:deny
everyone@: r-x---a-R-c--s:-----:allow

```

Une autre méthode d'ajout des mêmes droits pour l'utilisateur gozer consiste à insérer une ACL à un emplacement spécifique, par exemple 4. Ainsi, les ACL existantes aux emplacements 4-6 sont déplacées vers le bas. Exemple :

```

# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r-xr-x+ 1 root    root        206663 Feb 16 11:00 file.1
  owner@: --x-----:-----:deny
  owner@: rw-p---A-W-Co-:-----:allow
  group@: -w-p-----:-----:deny
  group@: r-x-----:-----:allow
  user:gozer:r-x-----:-----:allow
  everyone@: -w-p---A-W-Co-:-----:deny
  everyone@: r-x---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:rw:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+ 2 root    root            2 Aug 28 13:21 dir.2
  user:gozer:rw-----:fd----:allow
  owner@: -----:-----:deny
  owner@: rwxp---A-W-Co-:-----:allow
  group@: -w-p-----:-----:deny
  group@: r-x-----:-----:allow
  everyone@: -w-p---A-W-Co-:-----:deny
  everyone@: r-x---a-R-c--s:-----:allow

```

Vous pouvez également couper et coller les droits et les indicateurs d'héritage à partir de la sortie `ls -V` en format `chmod` compact. Par exemple, afin de dupliquer les droits et les indicateurs d'héritage du fichier `dir.2` de l'utilisateur gozer à l'utilisateur cindys dans le fichier `dir.2`, copiez et collez les droits et les indicateurs d'héritage (`rw-----:f-----:allow`) dans la commande `chmod`. Exemple :

```

# chmod A+user:cindys:rw-----:fd----:allow dir.2
# ls -dV dir.2
drwxr-xr-x+ 2 root    root            2 Aug 28 14:12 dir.2
  user:cindys:rw-----:fd----:allow

```

EXEMPLE 8-11 Configuration et affichage des ACL en format compact (Suite)

```
user:gozer:rwx-----:fd----:allow
owner@:-----:-----:deny
owner@:rwxp---A-W-Co-:-----:allow
group@:-w-p-----:-----:deny
group@:r-x-----:-----:allow
everyone@:-w-p---A-W-Co-:-----:deny
everyone@:r-x---a-R-c--s:-----:allow
```


Administration déléguée de ZFS

Ce chapitre décrit la méthode d'utilisation de l'administration déléguée pour permettre aux utilisateurs ne disposant pas de privilèges appropriés d'effectuer des tâches d'administration ZFS.

- “Présentation de l'administration déléguée de ZFS” à la page 221
- “Délégation de droits ZFS” à la page 222
- “Affichage des droits ZFS délégués (exemples)” à la page 226
- “Délégation de droits ZFS (exemples)” à la page 227
- “Suppression de droits ZFS (exemples)” à la page 232

Présentation de l'administration déléguée de ZFS

Cette fonction vous permet de distribuer des droits précis à des utilisateurs ou des groupes spécifiques, voire à tous les utilisateurs. Deux types de droits délégués sont pris en charge :

- Des droits individuels peuvent être explicitement spécifiés, notamment de création (create), de destruction (destroy), de montage (mount), d'instantané (snapshot), etc.
- Des groupes de droits appelés *jeux de droits* peuvent être définis. Tout utilisateur d'un jeu de droits est automatiquement affecté par les modifications apportées à celui-ci dans le cadre d'une mise à jour. Les jeux de droits commencent par la lettre @ et sont limités à 64 caractères. Les caractères suivant le caractère @ 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. Le modèle de délégation ZFS offre les avantages suivants pour la gestion des pools de stockage et systèmes de fichiers ZFS :

- Les droits sont transférés avec le pool de stockage ZFS lorsque celui-ci est migré.
- Offre un héritage dynamique vous permettant de contrôler la propagation des droits dans les systèmes de fichiers.

- Peut être configuré de manière à ce que seul le créateur d'un système de fichiers puisse détruire celui-ci.
- Les droits peuvent être distribués à des systèmes de fichiers spécifiques. Tout nouveau système de fichiers peut automatiquement récupérer des droits.
- Ce modèle offre une administration NFS simple. Un utilisateur disposant de droits 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 de Solaris, reportez-vous à [Partie III, “Roles, Rights Profiles, and Privileges”](#) du *System Administration Guide: Security Services* (en anglais).

Désactivation des droits délégués de ZFS

Vous pouvez activer ou désactiver l'administration déléguée en définissant la propriété `delegation` du pool. 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 de droits ZFS

Vous pouvez utiliser la commande `zfs allow` pour accorder des droits applicables aux jeux de données ZFS aux utilisateurs non root, de la manière suivante :

- Vous pouvez accorder des droits individuels à un utilisateur, à un groupe, voire à tous les utilisateurs.
- Vous pouvez accorder des groupes de droits individuels sous forme de *jeu de droits* à un utilisateur, à un groupe, voire à tous les utilisateurs.
- Vous pouvez accorder des droits localement uniquement au jeu de données actuel ou à tous les descendants de celui-ci.

Le tableau suivant décrit les opérations pouvant être déléguées et tout droit dépendant devant réaliser ces opérations déléguées.

Droit (sous-commande)	Description	Dépendances
allow	Capacité à accorder des droits qui vous ont été octroyés à un autre utilisateur.	Doit également disposer du droit à autoriser.
Clone	Capacité à cloner tout instantané du jeu de données.	Doit également disposer de la capacité create et de la capacité mount dans le système de fichiers d'origine.
create	Capacité à créer des jeux de données descendants.	Doit également disposer de la capacité mount.
destroy	Capacité à détruire un jeu de données.	Doit également disposer de la capacité mount.
monter	Capacité à monter et démonter un jeu de données, et à créer et détruire les liens vers des périphériques de volume.	
promote	Capacité à promouvoir le clonage d'un jeu de données.	Doit également disposer de la capacité mount et de la capacité promote dans le système de fichiers d'origine.
receive	Capacité à créer un système de fichiers descendant à l'aide de la commande zfs receive.	Doit également disposer de la capacité mount et de la capacité create.
rename	Capacité à renommer un jeu de données.	Doit également disposer de la capacité create et de la capacité mount dans le nouveau parent.
rollback	Capacité à restaurer un instantané.	Doit également disposer de la capacité mount.
send	Capacité à envoyer un flux d'instantané.	
share	Capacité à partager et annuler le partage d'un-jeu de données.	
Instantané	Capacité à prendre un instantané de-jeu de données.	

Vous pouvez en outre déléguer les propriétés ZFS suivantes à des utilisateurs non root :

- aclinherit
- aclmode
- atime
- canmount
- casesensitivity
- Somme de contrôle
- compression
- copies

- devices
- exec
- mountpoint
- nbmand
- normalization
- quota
- readonly
- recordsize
- reservation
- setuid
- shareiscsi
- sharenfs
- sharesmb
- snapdir
- userprop
- utf8only
- version
- 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 143.

Description de la syntaxe de la délégation de droits (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 droits sont délégués :

```
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` accorde des droits `create-time`.

La syntaxe de `zfs allow` suivante (en gras) identifie la méthode de spécification des droits et jeux de droits :

```
zfs allow [-s] ... perm|@setname [...] filesystem | volume
```

Vous pouvez spécifier plusieurs droits sous forme de liste séparée par des virgules. Les noms de droits sont identiques aux sous-commandes et propriétés ZFS. Pour plus d'informations, reportez-vous à la section précédente.

Les droits peuvent être regroupés en *jeux de droits* et sont identifiés par l'option `-s`. Les jeux de droits peuvent être utilisés par d'autres commandes `zfs allow` pour le système de fichiers spécifié et ses descendants. Les jeux de droits sont évalués dynamiquement et de ce fait, toute modification apportée à un jeu est immédiatement mise à jour. Les jeux de droits doivent se conformer aux mêmes conventions 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 droits :

```
zfs allow [-ld] ... .. filesystem | volume
```

L'option `-l` indique que le droit est accordé au jeu de données spécifié mais pas à ses descendants, à moins de spécifier également l'option `-d`. L'option `-d` indique que le droit est accordé pour les jeux de données descendants mais pas pour l'actuel jeu de données, à moins de spécifier également l'option `-l`. Si aucun des droits `-ld` n'est spécifié, les droits sont accordés au système de fichiers ou au volume, ainsi qu'à leurs descendants.

Suppression des droits délégués de ZFS (`zfs unallow`)

Vous pouvez supprimer des droits précédemment accordés, à l'aide de la commande `zfs unallow`.

Supposons par exemple que vous déléguez les droits `create`, `destroy`, `mount` et `snapshot` de la manière suivante :

```
# zfs allow cindys create,destroy,mount,snapshot tank/cindys
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
      user cindys create,destroy,mount,snapshot
-----
```

Pour supprimer ces droits, vous devez utiliser une syntaxe du type suivant :

```
# zfs unallow cindys tank/cindys
# zfs allow tank/cindys
```

Utilisation de l'administration déléguée de ZFS

Cette section contient des exemples d'affichage et de délégation de droits ZFS délégués.

Affichage des droits ZFS délégués (exemples)

Vous pouvez vous servir de la commande suivante pour afficher les droits :

```
# zfs allow dataset
```

Cette commande affiche les droits définis ou accordés à ce jeu de données. La sortie contient les composants suivants :

- Jeux de droits
- Droits spécifiques ou droits à la création
- Jeu de données local
- Jeux de données locaux et descendants
- Jeux de données descendants uniquement

EXEMPLE 9-1 Affichage des droits d'administration déléguée de base

La sortie suivante de cet exemple indique que l'utilisateur `cindys` dispose des droits de création, de destruction, de montage et d'instantané dans le système de fichiers `tank/cindys`.

```
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
  user cindys create,destroy,mount,snapshot
```

EXEMPLE 9-2 Affichage des droits d'administration déléguée complexes

La sortie de cet exemple indique les droits suivants sur les systèmes de fichiers `pool/red` et `pool`.

Pour le système de fichiers `pool/red` :

- Deux jeux de droits sont définis :
 - `@eng` (create, destroy, snapshot, mount, clone, promote, rename)
 - `@simple` (create, mount)
- Les droits à la création sont définis pour le jeu de droits `@eng` et la propriété `mountpoint`. "À la création" signifie qu'une fois qu'un jeu de données est créé, le jeu de droits `@eng` et la propriété `mountpoint` sont accordés.
- Le jeu de droits `@eng` est accordé à l'utilisateur `tom` et les droits `create`, `destroy` et `mount` pour les systèmes de fichiers locaux sont accordés à l'utilisateur `joe`.

EXEMPLE 9-2 Affichage des droits d'administration déléguée complexes (Suite)

- Le jeu de droits `@basic`, ainsi que les droits `share` et `rename` pour les systèmes de fichiers locaux et descendants sont accordés à l'utilisateur `fred`.
- Le jeu de droits `@basic` pour les systèmes de fichiers descendants uniquement est accordé à l'utilisateur `barney` et au groupe `staff`.

Pour le système de fichiers `pool` :

- Le jeu de droits `@simple` (`create`, `destroy`, `mount`) est défini.
- Le jeu de droits sur le système de fichiers local `@simple` est accordé au groupe `staff`.

La sortie de cet exemple est la suivante :

```
$ zfs allow pool/fred
-----
Permission sets on (pool/fred)
    @eng create,destroy,snapshot,mount,clone,promote,rename
    @simple create,mount
Create time permissions on (pool/fred)
    @eng,mountpoint
Local permissions on (pool/fred)
    user tom @eng
    user joe create,destroy,mount
Local+Descendent permissions on (pool/fred)
    user fred @basic,share,rename
Descendent permissions on (pool/fred)
    user barney @basic
    group staff @basic
-----
Permission sets on (pool)
    @simple create,destroy,mount
Local permissions on (pool)
    group staff @simple
-----
```

Délégation de droits ZFS (exemples)

EXEMPLE 9-3 Délégation de droits à un utilisateur individuel

Lorsque vous accordez les droits `create` et `mount` à un utilisateur individuel, vous devez vous assurer que cet utilisateur dispose de droits sur le point de montage sous-jacent.

Pour accorder par exemple à l'utilisateur `marks` les droits `create` et `mount` sur `tank`, définissez au préalable ces droits :

```
# chmod A+user:marks:add_subdirectory:fd:allow /tank
```

EXEMPLE 9-3 Délégation de droits à un utilisateur individuel (Suite)

Utilisez ensuite la commande `zfs allow` pour accorder les droits `create`, `destroy` et `mount`.
Exemple :

```
# zfs allow marks create,destroy,mount tank
```

L'utilisateur `marks` peut dorénavant créer ses propres systèmes de fichiers dans le système de fichiers `tank`. Exemple :

```
# su marks
marks$ zfs create tank/marks
marks$ ^D
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied
```

EXEMPLE 9-4 Délégation des droits de création (`create`) et de destruction (`destroy`) à un groupe

L'exemple suivant illustre la configuration d'un système de fichiers pour que tout membre du groupe `staff` puisse créer et monter des systèmes de fichiers dans le système de fichiers `tank`, ainsi que 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
# zfs allow -c create,destroy tank
# zfs allow tank
-----
Create time permissions on (tank)
      create,destroy
Local+Descendent permissions on (tank)
      group staff create,mount
-----
# su cindys
cindys% zfs create tank/cindys
cindys% exit
# su marks
marks% zfs create tank/marks/data
marks% exit
cindys% zfs destroy tank/marks/data
cannot destroy 'tank/mark': permission denied
```

EXEMPLE 9-5 Délégation de droits au niveau approprié d'un système de fichiers

Assurez-vous d'accorder les droits aux utilisateurs au niveau approprié du système de fichiers. Par exemple, les droits `create`, `destroy` et `mount` pour les systèmes de fichiers locaux et

EXEMPLE 9-5 Délégation de droits au niveau approprié d'un système de fichiers (Suite)

descendants sont accordés à l'utilisateur `marks`. Le droit local de prendre un instantané du système de fichiers `tank` a été accordé à l'utilisateur `marks`, mais pas celui de prendre un instantané de son propre système de fichiers. Le droit `snapshot` ne lui donc a pas été accordé au niveau approprié du système de fichiers.

```
# zfs allow -l marks snapshot tank
# zfs allow tank
-----
Local permissions on (tank)
    user marks snapshot
Local+Descendent permissions on (tank)
    user marks create,destroy,mount
-----
# su marks
marks$ zfs snapshot tank/@snap1
marks$ zfs snapshot tank/marks@snap1
cannot create snapshot 'mark/marks@snap1': permission denied
```

Pour accorder à l'utilisateur `marks` ce droit au niveau des descendants, utilisez l'option `zfs allow -d`. Exemple :

```
# zfs unallow -l marks snapshot tank
# zfs allow -d marks snapshot tank
# zfs allow tank
-----
Descendent permissions on (tank)
    user marks snapshot
Local+Descendent permissions on (tank)
    user marks create,destroy,mount
-----
# su marks
$ zfs snapshot tank@snap2
cannot create snapshot 'sandbox@snap2': permission denied
$ zfs snapshot tank/marks@snappy
```

L'utilisateur `marks` ne peut maintenant créer un instantané qu'à un niveau inférieur à `tank`.

EXEMPLE 9-6 Définition et utilisation de droits délégués complexes

Vous pouvez accorder des droits spécifiques à des utilisateurs ou des groupes. Par exemple, la commande `zfs allow` suivante accorde des droits spécifiques au groupe `staff`. En outre, les droits `destroy` et `snapshot` sont accordés après la création de systèmes de fichiers `tank`.

```
# zfs allow staff create,mount tank
# zfs allow -c destroy,snapshot tank
```

EXEMPLE 9-6 Définition et utilisation de droits délégués complexes (Suite)

```
# zfs allow tank
-----
Create time permissions on (tank)
    destroy,snapshot
Local+Descendent permissions on (tank)
    group staff create,mount
-----
```

Étant donné que l'utilisateur `marks` est membre du groupe `staff`, il peut créer des systèmes de fichiers dans `tank`. En outre, l'utilisateur `marks` peut créer un instantané de `tank/marks2` parce qu'il dispose des droits spécifiques pour le faire. Exemple :

```
# su marks
$ zfs create tank/marks2
$ zfs allow tank/marks2
-----
Local permissions on (tank/marks2)
    user marks destroy,snapshot
-----
Create time permissions on (tank)
    destroy,snapshot
Local+Descendent permissions on (tank)
    group staff create
    everyone mount
-----
```

Il ne peut par contre pas créer d'instantané dans `tank/marks` parce qu'il ne dispose pas des droits spécifiques pour le faire. Exemple :

```
$ zfs snapshot tank/marks2@snap1
$ zfs snapshot tank/marks@snapp
cannot create snapshot 'tank/marks@snapp': permission denied
```

Si vous disposez des droits `create` dans votre répertoire personnel, vous pouvez créer vos propres répertoires d'instantanés. Ce scénario s'avère utile lorsque votre système de fichiers est monté sur un système NFS. Exemple :

```
$ cd /tank/marks2
$ ls
$ cd .zfs
$ ls
snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x  2 marks  staff          2 Dec 15 13:53 snap1
```

EXEMPLE 9-6 Définition et utilisation de droits délégués complexes (Suite)

```

$ pwd
/tank/marks2/.zfs/snapshot
$ mkdir snap2
$ zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
tank                                264K  33.2G  33.5K  /tank
tank/marks                          24.5K  33.2G  24.5K  /tank/marks
tank/marks2                          46K   33.2G  24.5K  /tank/marks2
tank/marks2@snap1                   21.5K   -    24.5K  -
tank/marks2@snap2                    0     -    24.5K  -
$ ls
snap1  snap2
$ rmdir snap2
$ ls
snap1

```

EXEMPLE 9-7 Définition et utilisation d'un jeu de droits délégué ZFS

L'exemple suivant illustre la création d'un jeu de droits intitulé `@myset` et, accorde ce jeu de droits ainsi que le droit de renommage au groupe `staff` pour le système de fichiers `tank`. L'utilisateur `cindys`, membre du groupe `staff`, a le droit de créer un système de fichiers dans `tank`. Par contre, l'utilisateur `lp` ne dispose pas de ce droit de création de systèmes de fichiers dans `tank`.

```

# zfs allow -s @myset create,destroy,mount,snapshot,promote,clone,readonly tank
# zfs allow tank
-----
Permission sets on (tank)
      @myset clone,create,destroy,mount,promote,readonly,snapshot
-----
# zfs allow staff @myset,rename tank
# zfs allow tank
-----
Permission sets on (tank)
      @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions on (tank)
      group staff @myset,rename
# chmod A+group:staff:add_subdirectory:fd:allow tank
# su cindys
cindys% zfs create tank/data
Cindys% zfs allow tank
-----
Permission sets on (tank)
      @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions on (tank)

```

EXEMPLE 9-7 Définition et utilisation d'un jeu de droits délégué ZFS (Suite)

```

group staff @myset, rename
-----
cindys% ls -l /tank
total 15
drwxr-xr-x  2 cindys  staff          2 Aug  8 14:10 data
cindys% exit
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied

```

Suppression de droits ZFS (exemples)

Vous pouvez utiliser la commande `zfs unallow` pour supprimer des droits accordés. Par exemple, l'utilisateur `cindys` a le droit de créer, détruire, monter et réaliser des instantanés dans le système de fichiers `tank/cindys`.

```

# zfs allow cindys create,destroy,mount,snapshot tank/cindys
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
      user cindys create,destroy,mount,snapshot
-----

```

La syntaxe suivante de la commande `zfs unallow` supprime le droit de réaliser des instantanés du système de fichiers `tank/cindys` accordé à l'utilisateur `cindys` :

```

# zfs unallow cindys snapshot tank/cindys
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
      user cindys create,destroy,mount
-----
cindys% zfs create tank/cindys/data
cindys% zfs snapshot tank/cindys@today
cannot create snapshot 'tank/cindys@today': permission denied

```

Autre exemple : l'utilisateur `marks` dispose des droits suivants dans `tank/marks` :

```

# zfs allow tank/marks
-----
Local+Descendent permissions on (tank/marks)
      user marks create,destroy,mount
-----

```

Dans cet exemple, la syntaxe suivante de la commande `zfs unallow` supprime tous les droits accordés à l'utilisateur `marks` pour `tank/marks`:

```
# zfs unallow marks tank/marks
```

La syntaxe suivante de la commande `zfs unallow` supprime un jeu de droits sur le système de fichiers `tank`.

```
# zfs allow tank
```

```
-----
Permission sets on (tank)
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Create time permissions on (tank)
    create,destroy,mount
Local+Descendent permissions on (tank)
    group staff create,mount
-----
# zfs unallow -s @myset tank
$ zfs allow tank
-----
Create time permissions on (tank)
    create,destroy,mount
Local+Descendent permissions on (tank)
    group staff create,mount
-----
```


Sections avancées de ZFS

Ce chapitre décrit les volumes ZFS, l'utilisation de ZFS dans un système Solaris avec zones installées, les pools racine de remplacement ZFS et les profils de droits ZFS.

Il contient les sections suivantes :

- “Volumes ZFS” à la page 235
- “Utilisation de ZFS dans un système Solaris avec zones installées” à la page 238
- “Utilisation de pools racine ZFS de remplacement” à la page 244
- “Profils de droits ZFS” à la page 245

Volumes ZFS

Un volume ZFS est un jeu de données représentant un périphérique en mode bloc qui peut être utilisé en tant que tel. Les volumes ZFS sont identifiés en tant que périphériques dans le répertoire `/dev/zvol/{dsk, rdsk}/path`.

Dans l'exemple suivant, le volume ZFS `tank/vol` de 5 Go a été créé :

```
# zfs create -V 5gb tank/vol
```

Lors de la création d'un volume, une réservation est automatiquement configurée à la taille initiale du volume. La taille de la réservation est maintenue à égalité avec la taille du volume pour éviter tout comportement inattendu. Si, par exemple, la taille du volume diminue, les données risquent d'être corrompues. Vous devez faire preuve de prudence lors de la modification de la taille du volume.

En outre, en cas de création de l'instantané d'un volume dont la taille change, cela peut entraîner des incohérences dans le système de fichiers si vous tentez un retour arrière de 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 6-1](#).

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. Toute tentative de création ou de clonage d'un volume à partir d'une zone non globale échoue. 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 240.

Utilisation d'un volume ZFS en tant que périphérique de swap ou de dump

Lors de l'installation d'un système de fichiers racine ZFS ou d'une migration à partir d'un système de fichiers UFS, un périphérique de swap est créé sur un volume ZFS du pool racine ZFS. La taille de la zone de swap est basée sur la moitié de la taille de la mémoire physique.

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 racine ZFS ou d'une migration à partir d'un système de fichiers UFS, un périphérique de vidage est créé sur un volume ZFS du pool racine ZFS. La taille du périphérique de vidage est basée sur la moitié de la taille de la mémoire physique. Le périphérique de vidage ne nécessite aucune administration une fois configuré.

Exemple :

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
Savecore enabled: yes
```

En raison de CR 6724860, vous devez exécuter la commande `savecore` manuellement pour enregistrer un vidage mémoire sur incident lorsque vous utilisez un volume de vidage ZFS.

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 Solaris précédentes. Pour définir une zone de swap supplémentaire, créez un volume ZFS d'une taille spécifique et activez le swap sur ce périphérique.

Pour définir une zone de swap, créez un volume ZFS d'une taille spécifique et activez le swap dans ce périphérique.

Dans l'exemple suivant, le volume `tank/vol` de 5 Go est ajouté en tant que périphérique de swap.

```
# zfs create -V 5gb tank/vol
# swap -a /dev/zvol/dsk/tank/vol
# swap -l
swapfile                dev  swaplo  blocks  free
/dev/dsk/c0t0d0s1       32,33  16 1048688 1048688
/dev/zvol/dsk/tank/vol  254,1   16 10485744 10485744
```

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 82.

Utilisation d'un volume ZFS en tant que cible iSCSI Solaris

Les initiateurs et les cibles iSCSI Solaris sont pris en charge dans cette version de Solaris.

De plus, la création d'un volume ZFS en tant que cible iSCSI s'effectue facilement, en configurant la propriété `shareiscsi` dans le volume. 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 14, “Configuring Solaris iSCSI Targets and Initiators \(Tasks\)”](#) du *System Administration Guide: Devices and File Systems*.

Remarque – La commande `iscsitadm` permet la création et la gestion de cibles iSCSI. Si vous avez configuré la propriété `shareiscsi` dans un volume ZFS, n'utilisez pas la commande `iscsitadm` pour créer le même périphérique cible. Dans le cas contraire, vous obtiendrez des informations de cible dupliquées pour le même périphérique.

La gestion d'un volume ZFS en tant que cible iSCSI s'effectue comme pour tout 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. 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. 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 de ZFS dans un système avec zones Solaris.

- [“Ajout de systèmes de fichiers ZFS à une zone non globale” à la page 239](#)
- [“Délégation de jeux de données à une zone non globale” à la page 240](#)
- [“Ajout de volumes ZFS à une zone non globale” à la page 240](#)
- [“Utilisation de pools de stockage ZFS au sein d'une zone” à la page 241](#)
- [“Gestion de propriétés ZFS au sein d'une zone” à la page 241](#)
- [“Explication de la propriété zoned” à la page 242](#)

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 ZFS ou un clone ZFS à une zone non globale en déléguant ou non le contrôle administratif.
- Il est possible d'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 ci-dessous, un 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 modifier les jeux de données non ajoutés à la zone et ne peut pas dépasser les quotas maximum définis pour les jeux de données exportés.

Tenez compte des interactions suivantes lorsque vous travaillez avec ZFS dans un système ZFS sur lequel des zones Solaris sont installées :

- La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`.
- En raison du bogue 6449301, 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.

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`.

La sous-commande `add fs` de la commande `zonecfg` permet d'ajouter un système de fichiers ZFS à une zone non globale. Exemple :

Dans l'exemple suivant, un système de fichiers ZFS est ajouté à une zone non globale par un administrateur global dans 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=/export/shared
zonecfg:zion:fs> end
```

Cette syntaxe permet d'ajouter le système de fichiers ZFS `tank/zone/zion` à la zone `zion` déjà configurée, montée à `/export/shared`. 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 emplacement différent. En outre,

l'administrateur de zone ne peut pas modifier les propriétés dans le système de fichiers, telles que 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 sur la configuration des types de ressources à l'aide de `zonecfg`, reportez-vous à la [Partie II, "Zones" du Guide d'administration système : Gestion des ressources conteneurs Solaris et des zones Solaris](#).

Délégation de jeux de données à une zone non globale

Si le principal objectif est de déléguer l'administration du stockage à une zone, ZFS prend alors 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 enfants. En outre, il peut prendre des instantanés, créer des clones et contrôler la hiérarchie complète du système de fichiers.

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 241.

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`. En cas de détection d'une tentative d'ajout d'un volume ZFS, la zone ne peut pas se réinitialiser. 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 dans 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 exporte le volume `tank/vol` dans la zone. 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 des zones et les risques de sécurité associés, reportez-vous à la section [“Explication de la propriété `zoned`”](#) à la page 242.

Pour plus d'informations sur l'ajout de périphériques à des zones, reportez-vous à la [Partie II, “Zones” du Guide d'administration système : Gestion des ressources conteneurs Solaris et des zones Solaris](#).

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 l'ajout d'un jeu de données à une zone, l'administrateur de zone peut contrôler des propriétés de jeux de données spécifiques. Lorsqu'un jeu de données est ajouté à une zone, tous ses ancêtres sont visibles en tant que jeux de données en lecture seule, tandis que le jeu de données lui-même est accessible en écriture, de même que ses enfants. 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
<code>tank</code>	Oui	Non	-
<code>tank/home</code>	Non	-	-
<code>tank/data</code>	Oui	Non	-
<code>tank/data/matrix</code>	Non	-	-
<code>tank/data/zion</code>	Oui	Oui	<code>sharenfs, zoned, quota, reservation</code>
<code>tank/data/zion/home</code>	Oui	Oui	<code>sharenfs, zoned</code>

Notez que chaque parent de `tank/zone/zion` est visible en lecture seule, que tous les enfants sont accessibles en écriture et 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 NFS. Il ne peut pas non plus modifier la propriété `zoned` car cela entraînerait un risque de sécurité, tel que décrit dans la section suivante.

Toute autre propriété configurable peut être modifiée, à l'exception de la propriété `quota` et du jeu de données lui-même. Ce comportement permet à l'administrateur de zone globale de contrôler la consommation d'espace de l'ensemble des 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 l'ajout d'un jeu de données à une zone non globale.

Explication de la propriété `zoned`

Lors qu'un jeu de données est ajouté 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 ajouté à une zone non globale sous le contrôle d'un administrateur de zone, son contenu n'est plus fiable. Comme pour tout système 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 dans la zone globale est impossible et le jeu de données est ignoré lors de l'exécution de la commande `zfs share -a` ou de la commande `zfs mount -a`. Dans l'exemple suivant, `tank/zone/zion` a été ajouté à une zone, tandis que `tank/zone/global` ne l'a pas été :

```
# zfs list -o name,zoned,mountpoint -r tank/zone
NAME                ZONED  MOUNTPPOINT
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, pas 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 de prévenir les risques de sécurité accidentels, 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` pour le jeu de données et 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, il est possible de 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 racine ZFS de remplacement

Lors de sa création, un pool est intrinsèquement lié au système hôte. Le système hôte assure la maintenance des connaissances relatives au pool, ce qui lui permet de détecter l'indisponibilité du pool, le cas échéant. Bien que ces connaissances soient utiles aux opérations normales, elles peuvent causer des interférences lors de l'initialisation à partir d'autres supports ou lors de la création d'un pool dans un média amovible. La fonction de pool *racine de remplacement* de ZFS permet de résoudre ce problème. Un pool racine 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 à la racine du pool.

Création de pools racine de remplacement ZFS

Le plus souvent, la création d'un pool racine de remplacement s'effectue 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 racine de remplacement est créé à l'aide de l'option `-R`, le point de montage du système de fichiers racine est automatiquement défini sur `/`, qui est l'équivalent de la racine de remplacement elle-même.

Dans l'exemple suivant, un pool nommé `morpheus` est créé à l'aide `/mnt` en tant que chemin de racine 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 la racine 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 racine de remplacement. Ce système de fichiers peut ensuite être exporté ou importé sous un pool racine de remplacement arbitraire dans un système différent.

Importation de pools racine de remplacement

L'importation de pool s'effectue également à l'aide d'une racine 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 de la racine actuelle, mais sous un répertoire temporaire où pourront s'effectuer les réparations. Cette fonction peut également être utilisée lors du montage de médias amovibles comme décrit ci-dessus.

Dans l'exemple suivant, un pool nommé `morpheus` est importé à l'aide de `/mnt` en tant que chemin racine de remplacement : Cet exemple part du principe que `morpheus` a été précédemment exporté.

```
# zpool import -R /mnt morpheus
# zpool list morpheus
NAME                SIZE    USED    AVAIL    CAP    HEALTH    ALTROOT
morpheus            33.8G   68.0K   33.7G    0%    ONLINE    /mnt
# zfs list morpheus
NAME                USED    AVAIL    REFER    MOUNTPOINT
morpheus            32.5K   33.5G    8K      /mnt/morpheus
```

Profils de droits ZFS

Si vous souhaitez effectuer des tâches de gestion ZFS sans utiliser le compte superutilisateur (racine), vous pouvez adopter 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 ou manipuler les périphériques au sein d'un pool de stockage ZFS.
- Gestion de système de fichiers ZFS – Permet de créer, détruire et modifier les 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 [Chapitre9, “Administration déléguée de ZFS”](#).

Résolution de problèmes et récupération de données ZFS

Ce chapitre décrit les méthodes d'identification et de résolution des modes de panne de ZFS. Des informations relatives à la prévention des pannes sont également fournies.

Il contient les sections suivantes :

- “Modes de panne ZFS” à la page 247
- “Vérification de l'intégrité des données ZFS” à la page 249
- “Identification de problèmes dans ZFS” à la page 251
- “Réparation d'un configuration ZFS endommagée” à la page 257
- “Réparation d'un périphérique manquant” à la page 257
- “Réparation d'un périphérique endommagé” à la page 259
- “Réparation de données endommagées” à la page 267
- “Réparation d'un système impossible à réinitialiser” à la page 271

Modes de panne ZFS

En tant que système de fichiers et gestionnaire de volumes combinés, ZFS peut rencontrer différents modes de panne. Ce chapitre commence par une description des différents modes de panne, puis explique comment les identifier sur un système en cours d'exécution. Il se conclut en expliquant comment résoudre les problèmes. ZFS peut faire face à trois types d'erreurs de base :

- “Périphériques manquants dans un pool de stockage ZFS” à la page 248
- “Périphériques endommagés dans un pool de stockage ZFS” à la page 248
- “Données ZFS corrompue” à la page 248

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.

Périphériques manquants dans un pool de stockage ZFS

Si un périphérique est supprimé du système, ZFS détecte que son ouverture est impossible et le met dans l'état UNAVAIL. En fonction du niveau de réplication des données du pool, cela peut résulter ou pas 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. Si tous les composants d'un miroir sont supprimés, si plusieurs périphériques d'un périphérique RAID-Z sont supprimés, ou si un périphérique de niveau supérieur à un disque est supprimé, l'état du pool devient FAULTED. Aucune donnée n'est accessible tant que le périphérique n'est pas reconnecté.

Périphériques endommagés dans un pool de stockage ZFS

Le terme " endommagé " fait référence à une grande variété d'erreurs possibles. Les exemples incluent les erreurs suivantes :

- erreurs d'E/S transitoires causées par un disque ou un contrôleur défaillant ;
- corruption 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 ;
- écrasement accidentel de parties du périphérique physique par un autre 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 corruption 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 administrateur 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 de la séquence d'erreurs dans un périphérique n'est pas une tâche aisée. Elle est abordée plus en détail dans une section ultérieure.

Données ZFS corrompue

La corruption de données se produit lorsqu'une ou plusieurs erreurs de périphériques (indiquant des 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 de corruption de données. Si une erreur se produit sur l'autre côté du miroir au même emplacement, les données sont alors endommagées.

La corruption de données est toujours permanente et nécessite un 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 disques de routine, comme expliqué dans la section suivante. Lorsqu'un bloc corrompu est supprimé, le nettoyage de disque suivant reconnaît que la corruption n'est plus présente et supprime toute trace de l'erreur dans le système.

Vérification de l'intégrité des données ZFS

Il n'existe pas d'utilitaire `fsck` équivalent pour ZFS. Cet utilitaire remplissait deux fonctions : la réparation et la validation des données.

Réparation de données

Avec les systèmes de fichiers classiques, la méthode d'écriture des données est affectée par les pannes inattendues entraînant des incohérences de données. Un système de fichiers classique n'étant pas transactionnel, les blocs non référencés, les comptes de liens défectueux ou autres structures de données 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. Grâce à ZFS, ces problèmes ne se posent pas. Une incohérence des données sur disque 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` étant conçu pour réparer les problèmes spécifiques aux systèmes de fichiers individuels, l'écriture d'un tel utilitaire pour un système de fichiers ne présentant pas de problème connu est impossible. La preuve pourrait être apportée à l'avenir que certains problèmes de corruption de données sont suffisamment fréquents et simples pour justifier le développement d'un tel utilitaire de réparation, mais ces problèmes peuvent toujours être évités à l'aide de pools redondants.

Si le pool n'est pas redondant, le risque qu'une corruption de données puisse rendre tout ou partie de vos données inaccessibles est toujours présent.

Validation de données

Outre la réparation de données, l'utilitaire `fsck` valide l'absence de problème relatif aux données sur le disque. Cette tâche s'effectue habituellement en démontant le système de fichiers et en exécutant 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 à un utilitaire explicite pour effectuer la vérification nécessaire, ZFS fournit un mécanisme pour effectuer une vérification de routine des données. 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 le système de fichiers reste utilisable et sa réactivité est quasiment la même lors du nettoyage. La commande `zpool scrub` permet de lancer un nettoyage explicite. Exemple :

```
# zpool scrub tank
```

La sortie de `zpool status` permet d'afficher l'état du nettoyage actuel. Exemple :

```
# zpool status -v tank
pool: tank
state: ONLINE
scrub: scrub completed after 0h13m with 0 errors on Thu Aug 28 09:57:41 2008
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c1t0d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0

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

Notez qu'une seule opération de nettoyage actif par pool peut se produire à la fois.

L'option `-s` permet d'interrompre un nettoyage en cours. 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 également 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 123.

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 et le redémarre une fois qu'elle a terminé.

Pour de plus amples informations sur la réargenture, reportez-vous à la section [“Affichage de l'état de réargenture”](#) à la page 265.

Identification de problèmes dans ZFS

Les sections suivantes décrivent l'identification 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 253
- [“Consultation de la sortie de `zpool status`”](#) à la page 253
- [“Rapport système de messages d'erreur ZFS”](#) à la page 256

Les fonctions suivantes permettent d'identifier les problèmes au sein de la configuration ZFS :

- La commande `zpool status` permet d'obtenir des informations relatives au pool 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 plupart des solutions de problèmes ZFS sont axées autour de la commande `zpool status`. 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 de corruption de données impliquent toujours la panne d'un périphérique. Le remplacement d'un périphérique défaillant ne règle pas les problèmes de corruption de données.

En outre, un moteur de diagnostic ZFS est fourni pour diagnostiquer et rapporter les défaillances de pool ou de périphériques. Les erreurs liées aux sommes de contrôle, aux E/S et aux périphériques font également l'objet d'un rapport lorsqu'elles sont liées à la défaillance d'un pool ou d'un périphérique. Les défaillances ZFS telles que rapportées par `zpool status` s'affichent sur la console ainsi que dans le fichier de messages système. Dans la plupart des cas, le message `zpool status` 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 précédentes ayant entraîné le scénario d'erreur. Exemple :

```
# zpool history
History for 'tank':
2007-04-25.10:19:42 zpool create tank mirror c0t8d0 c0t9d0 c0t10d0
2007-04-25.10:19:45 zfs create tank/erick
2007-04-25.10:19:55 zfs set checksum=off tank/erick
```

Notez dans la sortie ci-dessous que les sommes de contrôle sont désactivées pour le système de fichiers `tank/erick`. Cette configuration est déconseillée.

- Identifiez les erreurs à l'aide des messages `zpool status` affichés sur la console système ou dans les fichiers `/var/adm/messages`.
- Obtenez des instructions de réparation supplémentaires grâce à la commande `zpool status -x`.
- Réparez les pannes, par exemple :
 - Remplacez le périphérique défaillant ou manquant et mettez-le en ligne.
 - Restaurez la configuration défaillante ou les données corrompues à partir d'une sauvegarde.
 - Vérifiez la récupération à l'aide de la commande `zpool status -x`.
 - Sauvegardez la configuration restaurée, le cas échéant.

Ce chapitre décrit les méthodes d'interprétation de la sortie de `zpool status` afin de diagnostiquer le type de panne et vous indique la section à consulter pour résoudre le problème. Bien que la commande effectue la plupart du travail de façon automatique, il est important de bien comprendre les problèmes identifiés afin de diagnostiquer le type de panne.

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. En l'absence de pools défectueux dans le système, la commande affiche un message simple, comme ci-dessous :

```
# 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 123.

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 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: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	DEGRADED	0	0	0
mirror	DEGRADED	0	0	0
c1t0d0	ONLINE	0	0	0
c1t1d0	OFFLINE	0	0	0

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

Cette sortie est divisée en plusieurs sections :

Informations globales d'état des pools

Cette section d'en-tête de la sortie `zpool status` se compose des champs suivants, certains d'entre eux n'étant affichés que pour les pools présentant des problèmes :

<code>pool</code>	Nom du pool.
<code>state</code>	État de maintenance actuel du pool. Ces informations concernent uniquement la capacité de pool à fournir le niveau de réplication requis. Les pools dont l'état est <code>ONLINE</code> peuvent contenir des périphériques défectueux ou des données corrompues.
<code>status</code>	Description du problème dans le pool. Ce champ est absent si aucun problème n'est détecté.
<code>action</code>	Action recommandée pour la réparation des erreurs. Ce champ constitue un formulaire abrégé qui redirige l'utilisateur vers l'une des sections suivantes. Ce champ est absent si aucun problème n'est détecté.
<code>see</code>	Référence à un article de connaissances contenant des informations de réparation détaillées. Les articles en ligne sont actualisés plus régulièrement que ce guide. Normalement, ils sont toujours référencés pour fournir les procédures de réparation les plus à jour. Ce champ est absent si aucun problème n'est détecté.
<code>scrub</code>	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 requête de nettoyage.
<code>errors</code>	Identifie les erreurs de données ou l'absence d'erreurs de données connus.

Informations de configuration

Le champ `config` de la sortie `zpool status` décrit la configuration des périphériques incluant 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`, `UNAVAILABLE` ou `OFFLINE`. 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` : des erreurs d'E/S se sont produites lors de l'envoi d'une requête de lecture.
- `WRITE` : des erreurs d'E/S se sont produites lors de l'envoi d'une requête d'écriture.
- `CKSUM` : erreurs de somme de contrôle. Le périphérique a renvoyé des données corrompues en réponse à une requête 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 la corruption de données telle qu'interprétée par les applications. Si la configuration du périphérique est redondante, les périphériques de disque peuvent présenter des erreurs impossibles à corriger, même si aucune erreur ne s'affiche au

niveau du périphérique RAID-Z ou du miroir. Si c'est ce scénario qui se produit, ZFS a récupéré les données adéquates et a réussi à réparer les données endommagées pour les répliques existantes.

Pour de plus amples informations sur l'interprétation de ces erreurs pour déterminer une panne de périphérique, reportez-vous à la section [“Détermination du type de panne de périphérique” à la page 259](#).

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 de modes de panne. Si l'état d'un périphérique est `FAULTED`, ce champ indique si le périphérique est inaccessible ou si les données du périphérique sont corrompues. Si le périphérique est en cours de réargenture, ce champ affiche la progression du processus.

Pour de plus amples informations sur le contrôle de la progression de la réargenture, reportez-vous à la section [“Affichage de l'état de réargenture” à la page 265](#).

État du nettoyage

La troisième section de la sortie `zpool status` décrit l'état actuel de tout 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 de corruption de données. Si le dernier nettoyage s'est récemment terminé, toute corruption de données existante aura probablement déjà été détecté.

Pour de plus amples informations sur le nettoyage de données et l'interprétation de ces informations, reportez-vous à la section [“Vérification de l'intégrité des données ZFS” à la page 249](#).

Erreurs de corruption de données

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 de disque ou lors des opérations normales. ZFS conserve un journal persistant des erreurs de données associées au pool. Ce journal tourne à chaque fois qu'un nettoyage complet du système est terminé.

Les erreurs de corruption 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 corrompues au sein du pool. Les erreurs de périphérique dans un pool redondant n'entraînent pas de corruption 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`. Exemple :

```
# zpool status -v
pool: tank
```

```

state: DEGRADED
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: resilver completed with 1 errors on Thu Aug 28 09:58:22 MDT 2008
config:

```

NAME	STATE	READ	WRITE	CKSUM	
tank	DEGRADED	0	0	1	
mirror	DEGRADED	0	0	1	
c1t0d0	ONLINE	0	0	2	
c1t1d0	UNAVAIL	0	0	0	corrupted data

errors: The following persistent errors have been detected:

DATASET	OBJECT	RANGE
5	0	lvl=4294967295 blkid=0

La commande `fmddump` affiche également un message similaire dans la console système et le fichier `/var/adm/messages`. La commande `fmddump` permet également de réaliser le suivi de ces messages.

Pour de plus amples informations sur l'interprétation d'erreurs de corruption de données, reportez-vous à la section [“Identification du type de corruption de données”](#) à la page 268.

Rapport système de messages d'erreur ZFS

Outre le suivi permanent des erreurs au sein du pool, ZFS affiche également des messages syslog lorsque des événements intéressants se produisent. Les scénarios suivants génèrent des événements pour notifier l'administrateur :

- **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.
- **Corruption de données** – En cas de détection de corruption 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 `fmddump`.

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.

Réparation d'un configuration ZFS endommagée

ZFS conserve un cache des pools actifs et de leur configuration dans le système de fichiers racine. Si ce fichier est corrompu ou n'est plus synchronisé avec ce qui est stocké dans le disque, l'ouverture du pool n'est plus possible. ZFS tente d'éviter ces situations, mais la corruption arbitraire reste possible 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 130.

Réparation d'un périphérique manquant

Si l'ouverture d'un périphérique est impossible, ce dernier s'affiche dans l'état UNAVAILABLE 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 refonctionner normalement.

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: Fri Aug 22 13:01:15 MDT 2008
PLATFORM: SUNW,Ultra-Enterprise, CSN: -, HOSTNAME: neo
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: 1f4f33d6-4973-4884-d494-a29b284d9554
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.

L'étape suivante consiste à exécuter la commande `zpool status -x` pour visualiser des informations plus détaillées sur le problème du périphérique et sa résolution. Exemple :

Cette sortie indique que le périphérique manquant `c0t1d0` ne fonctionne pas. Si vous déterminez que le périphérique est défectueux, remplacez-le.

Exécutez ensuite la commande `zpool online` pour mettre le périphérique remplacé en ligne. Exemple :

```
# zpool online tank c0t1d0
```

Confirmez que le pool dont le périphérique a été remplacé est fonctionnel.

```
# zpool status -x tank
pool 'tank' is healthy
```

Reconnexion physique du périphérique

La reconnexion d'un périphérique dépend du périphérique en question. S'il s'agit d'un disque connecté au réseau, la connectivité doit être restaurée. S'il s'agit d'un support USB ou autre support 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 (événement peu probable), le périphérique doit être traité comme un périphérique endommagé. Suivez les procédures décrites dans la section [“Réparation d'un périphérique endommagé” à la page 259](#).

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 défaillant ou si le system a été réinitialisé en tant que partie de la procédure de reconnexion, alors ZFS rebalayer 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 activé, 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`. 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 113.

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

Le terme *périphérique endommagé* peut décrire un grand nombre de situations :

- **Bit rot** – 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 dans des moments imprévisibles. Ces événements sont relativement rares mais, cependant, assez courants pour entraîner des corruptions de données potentielles dans des systèmes de grande taille ou de longue durée. Ces erreurs sont typiquement transitoires.
- **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 partie du disque, par exemple) qui entraînent la corruption 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. Il peut s'agir d'erreurs d'E/S constantes, de transports défectueux entraînant des corruptions aléatoires ou d'autres défaillances. 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. La première étape consiste à examiner le décompte d'erreurs dans la sortie de `zpool status` comme suit :

```
# zpool status -v pool
```

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, les problèmes générés par l'erreur d'administrateur peuvent entraîner un grand nombre d'erreurs. Le journal système 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 267.

Suppression des erreurs 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, les erreurs de périphériques 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`. Exemple :

```
# zpool clear tank c1t0d0
```

Cette syntaxe efface toutes les erreurs associées au 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, utilisez la syntaxe suivante :

```
# zpool clear tank
```

Pour de plus amples informations relatives à la suppression des erreurs de pool, reportez-vous à la section “[Suppression des périphériques de pool de stockage](#)” à la page 114.

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 261
- “[Périphériques impossibles à remplacer](#)” à la page 262
- “[Remplacement d'un périphérique dans un pool de stockage ZFS](#)” à la page 262
- “[Affichage de l'état de réargenture](#)” à la page 265

Détermination de la possibilité de remplacement du périphérique

Pour qu'un périphérique puisse être remplacé, l'état du pool doit être ONLINE. Le périphérique doit faire partie d'une configuration redondante ou être fonctionnel (état ONLINE). Si le disque 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 défectueux, chaque disque peut être remplacé car des répliques fonctionnelles sont disponibles. Cependant, en cas de panne de deux disques dans un périphérique RAID-Z à quatre directions, 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 FAULTED. 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 correcte, `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 défaillants ne peut être remplacé. Les disques `ONLINE` ne peuvent pas l'être non plus, car le pool lui-même est défaillant.

```
raidz          FAULTED
  c1t0d0       ONLINE
  c2t0d0       FAULTED
  c3t0d0       FAULTED
  c3t0d0       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 étaient défectueux, alors tout remplacement serait impossible car le pool lui-même serait défectueux.

Périphériques impossibles à remplacer

Si la perte d'un périphérique entraîne une défaillance du pool ou si le périphérique contient trop d'erreurs de données dans une configuration non redondante, alors 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 et de recréer la configuration, en restaurant les données au cours du processus.

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 270.

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 :

```
# zpool replace tank c1t0d0 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. Exemple :

```
# zpool replace tank c1t0d0
```

Cette commande formate adéquatement un disque non formaté et commence ensuite la réargenture de 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 114.

EXEMPLE 11-1 Remplacement d'un périphérique 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 Sun Fire x4500. Pour remplacer le disque `c1t3d0` par un nouveau au même emplacement (`c1t3d0`), annulez la configuration du disque avant de procéder au remplacement. Les étapes de base sont les suivantes :

- Mettez au préalable le disque à remplacer hors ligne. Vous ne pouvez pas annuler la configuration d'un disque utilisé.
- Identifiez le disque (`c1t3d0`) dont la configuration doit être annulée puis annulez sa configuration. Dans cette configuration mise 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, indiquant que le périphérique est prêt à être retiré, est allumée avant de retirer le lecteur défaillant.
- Reconfigurez le disque (`c1t3d0`).
- Remettez le 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=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.

```
# zpool offline tank c1t3d0
cfgadm | grep c1t3d0
sata1/3::disk/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
```

EXEMPLE 11-1 Remplacement d'un périphérique dans un pool de stockage ZFS (Suite)

```
sata1/3                disk            connected    unconfigured ok
<Replace the physical disk c1t3d0>
# cfgadm -c configure sata1/3
# cfgadm | grep sata3/7
sata3/7::disk/c5t7d0    disk            connected    configured   ok
# zpool online tank c1t3d0
# zpool replace tank c1t3d0
# zpool status
  pool: tank
  state: ONLINE
  scrub: resilver completed after 0h0m with 0 errors on Tue Apr 22 14:44:46 2008
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0t1d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0t2d0	ONLINE	0	0	0
c1t2d0	ONLINE	0	0	0
mirror	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*. 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` une fois le disque remplacé physiquement. Exemple :

```
# zpool replace tank c1t3d0 c4t3d0
# zpool status
  pool: tank
  state: DEGRADED
  scrub: resilver completed after 0h0m with 0 errors on Tue Apr 22 14:54:50 2008
```

EXEMPLE 11-1 Remplacement d'un périphérique dans un pool de stockage ZFS (Suite)

config:

NAME	STATE	READ	WRITE	CKSUM
tank	DEGRADED	0	0	0
mirror	ONLINE	0	0	0
c0t1d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0t2d0	ONLINE	0	0	0
c1t2d0	ONLINE	0	0	0
mirror	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 Apr 22 14:54:50 2008
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0t1d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0t2d0	ONLINE	0	0	0
c1t2d0	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0t3d0	ONLINE	0	0	0
c4t3d0	ONLINE	0	0	0

Affichage de l'état de réargenture

Le processus de remplacement d'un disque peut prendre beaucoup de temps, selon la taille du disque 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 contrôler 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 (et non pas du remplacement complet d'un périphérique), vous pouvez effectuer la réargenture du disque en quelques minutes ou quelques secondes, plutôt que d'effectuer la réargenture de l'intégralité du disque ou de compliquer l'opération avec la journalisation de " régions sales " prise en charge par certains gestionnaires de volume. 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. Exemple :

```
# zpool status tank
pool: tank
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 0h2m, 16.43% done, 0h13m to go
config:
      NAME                STATE      READ WRITE CKSUM
      tank                DEGRADED   0     0     0
        mirror
          replacing       DEGRADED   0     0     0
            c1t0d0        ONLINE    0     0     0
            c2t0d0        ONLINE    0     0     0
            c1t1d0        ONLINE    0     0     0
```

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 de remplacement (*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 le processus terminé, la nouvelle configuration complète s'applique, remplaçant l'ancienne configuration. Exemple :

```
# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h2m with 0 errors on Thu Aug 28 09:50:11 2008
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c2t0d0	ONLINE	0	0	0
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.

Réparation de données endommagées

Les sections suivantes décrivent comment identifier le type de corruption de données et comment réparer les données le cas échéant.

- [“Identification du type de corruption de données”](#) à la page 268
- [“Réparation d'un fichier ou répertoire corrompu”](#) à la page 269
- [“Réparation de dommages présents dans l'ensemble du pool de stockage ZFS”](#) à la page 270

ZFS utilise les sommes de contrôle, la redondance et les données autorétablissantes afin de réduire les risques de corruption de données. Cependant, la corruption de données peut se produire si le pool n'est pas redondant, si la corruption s'est produite alors que le pool était endommagé ou si une série d'événements improbables a corrompu plusieurs copies d'un élément de données. Quelle que soit la source, le résultat est le même : les données sont corrompues et par conséquent inaccessibles. Les actions à effectuer dépendent du type de données corrompue et de leurs valeurs relatives. Deux types de données peuvent être corrompus :

- **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 corrompues, le pool entier ou des parties complètes de la hiérarchie du jeu de données sont indisponibles.

- Données d'objet – Dans ce cas, la corruption 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 “[Vérification de l'intégrité des données ZFS](#)” à la page 249.

Identification du type de corruption de données

Par défaut, la commande `zpool status` indique qu'une corruption s'est produite, mais n'indique pas à quel endroit. Exemple :

```
# zpool status
  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: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
monkey	ONLINE	0	0	0
c1t1d0s6	ONLINE	0	0	0
c1t1d0s7	ONLINE	0	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 des circonstances normales, cette situation est vraie. Certaines interruptions de service temporaires peuvent entraîner une corruption 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 la corruption de données se produit dans des métadonnées au niveau du pool, la sortie est légèrement différente. 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 corruption 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 corrompu

En cas de corruption d'un fichier ou d'un répertoire, le système peut tout de même continuer à fonctionner, selon le type de corruption. Tout dommage est irréversible, à moins que des copies correctes des données n'existent sur le système. Si les données sont importantes, vous n'avez pas d'autre choix que de restaurer les données affectées à partir d'une sauvegarde. Vous pouvez pourtant récupérer les données suite à corruption sans avoir à restaurer tout le 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. La commande `zpool status -v` permet d'afficher une liste de noms de fichiers comportant des erreurs persistantes. 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: none requested
config:

```

NAME	STATE	READ	WRITE	CKSUM
monkey	ONLINE	0	0	0
c1t1d0s6	ONLINE	0	0	0
c1t1d0s7	ONLINE	0	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 sortie précédente est 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. 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. 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. Exemple :

```
monkey/dnode:<0x0>
```

- En cas de corruption d'un MOS (Meta-Object Set, jeu de méta-objet), la balise spéciale `<metadata>` s'affiche, suivie du numéro de l'objet.

Si la corruption se situe au sein des métadonnées d'un répertoire ou d'un fichier, vous devez déplacer le fichier vers un autre emplacement. Vous pouvez déplacer en toute sécurité un fichier ou répertoire vers un emplacement moins pratique, ce qui permettra de restaurer l'objet d'origine sur place.

Réparation de dommages présents dans l'ensemble du pool de stockage ZFS

Si les dommages sont présents dans les métadonnées du pool, il est impossible d'ouvrir ce dernier. Vous devez alors restaurer le pool et l'ensemble de ses données à partir de sauvegardes. 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 `zpool status` pour pouvoir le recréer une fois le pool détruit. 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'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 de réinitialisation, 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 racine 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 la réinitialisation à partir d'une racine 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 ci-dessus.
- Utilisez la commande `fmddump -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 `fmddump`.

Index

A

Accès

- Instantané ZFS
(exemple), 180

ACL

- ACL dans un fichier ZFS
 - Description détaillée, 200
- ACL sur un fichier ZFS
 - Description détaillée, 199
- Configuration d'ACL dans un fichier ZFS (mode compact)
 - Exemple, 217
- Configuration d'héritage d'ACL dans un fichier ZFS (mode détaillé)
 - (exemple), 206
- Définition d'ACL sur un fichier ZFS (mode compact)
 - Description, 216
- Définition des ACL sur un fichier ZFS (mode détaillé)
 - Description, 200
- Définition sur les fichiers ZFS
 - Description, 198
- Description, 191
- Description de format, 193
- Différences avec les ACL POSIX-draft, 192
- Héritage d'ACL, 196
- Indicateurs d'héritage d'ACL, 196
- Mode de propriété `aclmode`, 197
- Mode de propriété d'ACL, 197
- Mode de propriété `aclinherit`, 197
- Modification d'une ACL insignifiante dans un fichier ZFS (mode détaillé)

- ACL, Modification d'une ACL insignifiante dans un fichier ZFS (mode détaillé) (*Suite*)
 - (Exemple), 202
 - Privilèges d'accès, 194
 - Restauration d'une ACL insignifiante sur un fichier ZFS (mode détaillé)
 - (exemple), 205
 - Type d'entrée, 194
- ACL, modèle de Solaris, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 51
- ACL NFSv4
 - Description de format, 193
 - Différences avec les ACL POSIX-draft, 192
 - Héritage d'ACL, 196
 - Indicateurs d'héritage d'ACL, 196
 - Mode de propriété d'ACL, 197
 - Modèle
 - Description, 191
- ACL POSIX-draft, Description, 192
- ACL Solaris
 - Description de format, 193
 - Différences avec les ACL POSIX-draft, 192
 - Héritage d'ACL, 196
 - Indicateurs d'héritage d'ACL, 196
 - Mode de propriété d'ACL, 197
 - Nouveau modèle
 - Description, 191
- Administration déléguée, présentation, 221
- Administration déléguée de ZFS, présentation, 221
- Administration simplifiée, Description, 36
- Affichage
 - Droits délégués (exemple), 226

Affichage (Suite)

- État de maintenance d'un pool de stockage ZFS
 - Exemple, 128
 - État détaillé du fonctionnement du pool de stockage ZFS
 - (exemple), 129
 - État fonctionnel d'un pool de stockage
 - Description, 127
 - history, commande, 27
 - Rapport syslog de messages d'erreur ZFS
 - Description, 256
 - Statistiques d'E/S à l'échelle du pool de stockage ZFS
 - Exemple, 125
 - Statistiques d'E/S de pool de stockage vdev ZFS
 - Exemple, 126
 - Statistiques d'E/S de pools de stockage ZFS
 - Description, 125
- Ajout**
- Disques, à une configuration RAID-Z
 - (exemple), 108
 - Périphérique à un pool de stockage ZFS (zpool add)
 - Exemple, 106
 - Périphérique de journalisation mis en miroir
 - (exemple), 108
 - Système de fichiers ZFS à une zone non globale
 - Exemple, 239
 - Volume ZFS à une zone non globale
 - (exemple), 240
- Ajustement, Périphériques de swap et de vidage, 82
- Annulation du partage
 - Système de fichiers ZFS
 - exemple, 170
- atime, propriété, Description, 144
- Autorétablissement, Description, 96
- available, propriété, Description, 145

B

- Bloc d'initialisation, Installation à l'aide de `installboot` et de `installgrub`, 84

C

- canmount, propriété
 - Description, 145
 - Description détaillée, 153
- checksum, propriété, Description, 145
- Clone
 - Création
 - Exemple, 182
 - Définition, 37
 - Destruction
 - Exemple, 183
 - Fonctions, 182
- Comportement d'espace saturé, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 51
- Composants, Pool de stockage ZFS, 91
- Composants de ZFS, Exigences d'attribution de noms, 40
- compression, propriété, Description, 146
- compressratio, propriété, Description, 146
- Comptabilisation de l'espace ZFS, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 50
- Configuration
 - ACL dans un fichier ZFS (mode compact)
 - Exemple, 217
- Configuration en miroir
 - Description, 94
 - Fonction de redondance, 94
 - Vue conceptuelle, 94
- Configuration RAID-Z
 - Double parité, description, 94
 - Exemple, 98
 - Parité simple, description, 94
 - Redondance, fonction, 94
 - Vue conceptuelle, 94
- Configuration RAID-Z, ajout de disques,
 - (exemple), 108
- Configuration requise, Installation et Live Upgrade, 55
- Connexion
 - Périphérique, à un pool de stockage ZFS (`zpool attach`)
 - Exemple, 109

Contrôle

- Intégrité des données ZFS, 249
- Validation des données (nettoyage), 250

copies, propriété, Description, 146

Création**Clone ZFS**

- Exemple, 182

Hierarchie d'un système de fichiers ZFS, 44

Instantané ZFS

- (exemple), 178

Pool de stockage, à l'aide d'un périphérique de journalisation (exemple), 99

Pool de stockage RAID-Z à double parité (`zpool create`)

- Exemple, 98

Pool de stockage RAID-Z à parité simple (`zpool create`)

- Exemple, 98

Pool de stockage ZFS

- Description, 97

Pool de stockage ZFS (`zpool create`)

- (exemple), 42, 97

Pool de stockage ZFS mis en miroir (`zpool create`)

- (exemple), 97

Pool racine de remplacement

- (exemple), 244

Système de fichiers ZFS, 46

- (exemple), 140

Description, 140

Système de fichiers ZFS de base (`zpool create`)

- (exemple), 42

Volume émulé en tant que périphérique de swap

- (exemple), 236

Volume ZFS

- (exemple), 235

creation, propriété, Description, 146

D**Définition**

ACL sur les fichiers ZFS

- Description, 198

ACL sur un fichier ZFS (mode compact)

- Description, 216

Définition (Suite)

ACL sur un fichier ZFS (mode détaillé)

- Description, 200

`atime`, propriété ZFS

- (exemple), 159

compression, propriété

- (exemple), 46

Héritage d'ACL dans un fichier ZFS (mode détaillé)

- (exemple), 206

mountpoint, propriété, 46

Point de montage hérité

- (exemple), 166

Point de montage ZFS (`zfs set mountpoint`)

- (exemple), 165

quota, propriété (exemple), 47

Quota d'un système de fichiers ZFS (`zfs set quota`)

- Exemple, 171

Quota ZFS

- (exemple), 159

Réservation de système de fichiers ZFS

- Exemple, 173

`sharefs`, propriété

- (exemple), 46

Délégation

Droit (exemple), 227

Jeu de données à une zone non globale

- (exemple), 240

Délégation de droits, `zfs allow`, 224

Délégation de droits à un groupe, (exemple), 228

Délégation de droits à un utilisateur individuel,

- (exemple), 227

Démontage

Système de fichiers ZFS

- (exemple), 168

Dépannage

Détection de problèmes éventuels (`zpool status -x`), 253

Déterminer si un périphérique peut être remplacé

- Description, 261

Périphérique manquant (défaillant), 248

Problème d'identification, 252

Remplacement d'un périphérique (`zpool replace`)

- (exemple), 266

Dépannage (Suite)

- Remplacement d'un périphérique manquant (exemple), 257
 - Réparation d'un fichier ou répertoire endommagé
 - Description, 269
 - Réparation d'un système qui ne peut être initialisé
 - Description, 271
 - Réparation d'une configuration ZFS endommagée, 257
 - Réparation de dommages au niveau d'un pool
 - Description, 270
- Destruction**
- Clone ZFS
 - Exemple, 183
 - Instantané ZFS (exemple), 179
 - Pool de stockage ZFS
 - Description, 97
 - Pool de stockage ZFS (`zpool destroy`) (exemple), 105
 - Système de fichiers ZFS (exemple), 141
 - Système de fichiers ZFS comportant des systèmes dépendants (exemple), 142
- Détection**
- Niveaux de réplication incohérents
 - Exemple, 103
 - Périphériques en cours d'utilisation
 - Exemple, 102
- Détermination**
- Type de panne de périphérique
 - Description, 259
- Déterminer**
- Remplacement d'un périphérique
 - Description, 261
- devices, propriété, Description, 146**

- Différences existant entre les systèmes de fichiers traditionnels et ZFS (Suite)
 - Nouveau modèle ACL de Solaris, 51
 - Disque, Composant de pool de stockage ZFS, 92
 - Disque entier, Composant de pool de stockage ZFS, 92
 - Disque hot spare
 - Création (exemple), 116
 - Description
 - Exemple, 116
- Données**
- Corruption identifiée (`zpool status -v`)
 - Exemple, 255
 - Endommagées, 248
 - Nettoyage
 - Exemple, 250
 - Réargenture
 - Description, 251
 - Réparation, 249
 - Validation (nettoyage), 250
- Données vérifiées par somme de contrôle, Description, 35**

E

- Effacement**
- Périphérique d'un pool de stockage ZFS (`zpool clear`)
 - Description, 114
- Enregistrement**
- Données d'un système de fichiers ZFS (`zfs send`) (exemple), 185
- Entrelacement dynamique**
- Description, 96
 - Fonction de pool de stockage, 96
- Envoi et réception**
- Données d'un système de fichiers ZFS
 - Description, 184
- Etiquette EFI**
- Description, 92
 - Interaction avec ZFS, 92
- exec, propriété, Description, 146**
- Exigences d'attribution de noms, Composants ZFS, 40
 - Exigences matérielles et logicielles, 41

- Exportation
 Pool de stockage ZFS
 (exemple), 131
- F**
- Fichiers, En tant que composants d'un pool de stockage ZFS, 94
- Fonctions de réplication de ZFS, Mise en miroir ou RAID-Z, 94
- G**
- Gestion d'un volume traditionnel, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 51
- Granularité d'un système de fichiers, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 49
- H**
- Héritage
 Propriété ZFS (`zfs inherit`)
 Description, 159
- Hierarchie d'un système de fichiers, Création, 44
- `history`, commande, `zpool history`, 27
- I**
- Identification
 Pool de stockage ZFS à importer (`zpool import -a`)
 (exemple), 132
- Stockage requis, 43
- Type de corruption de données (`zpool status -v`)
 Exemple, 268
- Importation
 Pool de stockage ZFS
 (exemple), 135
- Pool de stockage ZFS, à partir de répertoires alternatifs (`zpool import -d`)
- Importation, Pool de stockage ZFS, à partir de répertoires alternatifs (`zpool import -d`) (*Suite*)
 (exemple), 134
- Pool racine de remplacement
 (exemple), 244
- Initialisation
 Environnement d'initialisation ZFS avec `boot -L` et `boot -Z` sur un système SPARC, 85
- Système de fichiers racine, 83
- Installation
 Système de fichiers racine ZFS
 (installation initiale), 57
- Configuration requise, 55
- Fonctions, 54
- Installation JumpStart, 63
- Installation de blocs d'initialisation
`installboot` et `installgrup`
 (exemple), 84
- Installation initiale d'un système de fichiers racine ZFS,
 (exemple), 57
- Installation JumpStart
 Système de fichiers racine
 Exemples de profils, 63
- Problèmes, 66
- Instantané
 Accès
 (exemple), 180
- Comptabilisation de l'espace, 181
- Création
 (exemple), 178
- Définition, 39
- Destruction
 (exemple), 179
- Fonctions, 177
- Renommer
 Exemple, 179
- Restauration
 Exemple, 181
- J**
- Jeu de données
 Définition, 38
- Description, 139

Jeu de droits défini, 221

Journal d'intention ZFS (ZIL), Description, 22

L

Lecture seule, propriétés ZFS, `mounted`, 147

Liste

Descendants d'un système de fichiers ZFS
(exemple), 156

Informations sur le pool ZFS, 44

Pool de stockage ZFS

Description, 123

Pools de stockage ZFS

Exemple, 124

Propriété ZFS (`zfs list`)

(exemple), 160

Propriété ZFS de script

(exemple), 163

Propriété ZFS par valeur source

(exemple), 162

Système de fichiers ZFS

(exemple), 156

Système de fichiers ZFS (`zfs list`)

(exemple), 47

Système de fichiers ZFS sans l'en-tête

Exemple, 158

Types de systèmes de fichiers ZFS

Exemple, 158

`luactivate`

Système de fichier racine

(exemple), 72

`lucreate`

Environnement d'initialisation ZFS à partir d'un
environnement d'initialisation ZFS

(exemple), 73

Migration d'un système de fichiers racine

(exemple), 70

M

Migration

Système de fichiers racine UFS vers système de
fichiers racine ZFS

Migration, Système de fichiers racine UFS vers système
de fichiers racine ZFS (*Suite*)

(Solaris Live Upgrade), 67

Système de fichiers racine UFS vers un système de
fichiers racine ZFS

Problèmes, 69

Migration d'un pool de stockage ZFS, Description, 130

Miroir, Définition, 38

Mise à niveau

Pool de stockage ZFS

Description, 137

Mise en ligne d'un périphérique

Pool de stockage ZFS (`zpool online`)

(exemple), 113

Mise en ligne et hors ligne de périphériques

Pool de stockage ZFS

Description, 111

Mise hors ligne d'un périphérique (`zpool offline`)

Pool de stockage ZFS

(exemple), 112

Mode de panne, 247

Périphérique endommagé, 248

Mode de propriété `aclmode`, 197

Mode de propriété `aclinherit`, 197

Mode de propriétés d'ACL, `aclinherit`, 144

Mode panne

Données endommagées, 248

Périphérique manquant (défaillant), 248

Modes de propriétés d'ACL, `aclmode`, 144

Modification

ACL insignifiante dans un fichier ZFS (mode
détaillé)

(Exemple), 202

Montage

Système de fichiers ZFS

Exemple, 166

Montage d'un système de fichiers ZFS, Différences

existant entre les systèmes de fichiers traditionnels et
ZFS, 51

Mot clé de profil JumpStart, Système de fichiers racine

ZFS, 64

`mounted`, propriété, Description, 147

`mountpoint`, propriété, Description, 147

N

Nettoyage

Exemple, 250

Validation des données, 250

Niveaux de réplication incohérents

Détection

Exemple, 103

Notification

périphérique reconnecté dans ZFS(zpool onLine)

Exemple, 258

O

origin, propriété, Description, 147

P

Partage

Système de fichiers ZFS

Description, 169

Exemple, 169

Périphérique de journalisation mis en miroir, Création d'un pool (exemple), 99

Périphérique de journalisation mis en miroir, ajout, (exemple), 108

Périphérique virtuel

Composant de pools de stockage ZFS, 100

Définition, 39

Périphériques de journalisation distincts, considérations, 22

Périphériques de swap et de vidage

Ajustement de la taille, 82

Description, 81

Points à prendre en compte, 81

Périphériques en cours d'utilisation

Détection

Exemple, 102

Point de montage

Automatique, 165

Gestion de ZFS

Description, 164

Héritage, 164

Par défaut pour les pools de stockage ZFS, 104

Point de montage (*Suite*)

Par défaut pour un système de fichiers ZFS, 140

Pool, Définition, 38

Pool de stockage mis en miroir (zpool create), (exemple), 97

Pool de stockage ZFS

Affichage de l'état de maintenance

Exemple, 128

Affichage de l'état détaillé du fonctionnement (exemple), 129

Affichage de l'état fonctionnel, 127

Affichage du processus de réargenture (exemple), 266

Ajout de périphérique (zpool add)

Exemple, 106

Composants, 91

Configuration en miroir, description, 94

Configuration RAID-Z, création (zpool create) Exemple, 98

Connexion de périphériques (zpool attach) Exemple, 109

Corruption de données identifiée (zpool status -v) Exemple, 255

Création (zpool create) (exemple), 97

Création d'une configuration mise en miroir (zpool create) (exemple), 97

Destruction (zpool destroy) (exemple), 105

Détection de problèmes éventuels (zpool status -x) Description, 253

Détermination du type de panne de périphérique Description, 259

Déterminer si un périphérique peut être remplacé Description, 261

Données endommagées Description, 248

Entrelacement dynamique, 96

Exportation (exemple), 131

Identification du type de corruption de données (zpool status -v)

- Pool de stockage ZFS, Identification du type de corruption de données (`zpool status -v`) (*Suite*)
 - Exemple, 268
 - Identification pour l'importation (`zpool import -a`) (exemple), 132
 - Importation (exemple), 135
 - Importation à partir de répertoires alternatifs (`zpool import -d`) (exemple), 134
 - Informations globales d'état des pools pour la résolution de problèmes
 - Description, 253
 - Liste
 - Exemple, 124
 - Message d'erreur système
 - Description, 256
 - Migration
 - Description, 130
 - Miroir
 - Définition, 38
 - Mise à niveau
 - Description, 137
 - Mise en ligne et hors ligne de périphériques
 - Description, 111
 - Mise hors ligne d'un périphérique (`zpool offline`) (exemple), 112
 - Mode de panne, 247
 - Nettoyage de données
 - Exemple, 250
 - Nettoyage des données
 - Description, 250
 - Nettoyage des données et réargenture
 - Description, 251
 - Notification d'un périphérique reconnecté dans ZFS (`zpool online`)
 - Exemple, 258
 - Périphérique endommagé
 - Description, 248
 - Périphérique manquant (défaillant)
 - Description, 248
 - Périphérique virtuel, 100
 - Point de montage par défaut, 104
- Pool de stockage ZFS (*Suite*)
 - Pool
 - Définition, 38
 - Pool racine de remplacement, 244
 - Problème d'identification
 - Description, 252
 - Profil de droits, 245
 - RAID-Z
 - Définition, 39
 - RAID-Z, description, 94
 - Réargenture
 - Définition, 39
 - Récupération d'un pool détruit
 - Exemple, 136
 - Remplacement d'un périphérique (`zpool replace`) (exemple), 114
 - Remplacement d'un périphérique manquant (exemple), 257
 - Remplacement de périphérique (`zpool replace`)
 - Exemple, 262
 - Réparation d'un fichier ou répertoire endommagé
 - Description, 269
 - Réparation d'un système qui ne peut être initialisé
 - Description, 271
 - Réparation d'une configuration ZFS endommagée, 257
 - Réparation de dommages au niveau d'un pool
 - Description, 270
 - Réparation des données
 - Description, 249
 - Script de sortie de pool de stockage
 - Exemple, 124
 - Séparation des périphériques (`zpool detach`)
 - Exemple, 111
 - Statistiques d'E/S à l'échelle du pool
 - Exemple, 125
 - Statistiques d'E/S vdev
 - Exemple, 126
 - Suppression d'un périphérique
 - Exemple, 114
 - Suppression des erreurs de périphérique (`zpool clear`)
 - Exemple, 260

- Pool de stockage ZFS (*Suite*)
 - Test (zpool create -n)
 - Exemple, 104
 - Utilisation de disques entiers, 92
 - Utilisation de fichiers, 94
 - Validation des données
 - Description, 250
- Pool de stockage ZFS (zpool online)
 - Mise en ligne d'un périphérique (exemple), 113
- Pool racine de remplacement
 - Création (exemple), 244
 - Description, 244
 - Importation (exemple), 244
- Profil de droits
 - Gestion de systèmes de fichiers et pools de stockage ZFS
 - Description, 245
- Propriété altroot, description, 121
- Propriété autoreplace, description, 121
- Propriété available, description, 121
- Propriété bootfs, description, 121
- Propriété capacity, description, 121
- Propriété de pool ZFS
 - altroot, 121
 - autoreplace, 121
 - available, 121
 - bootfs, 121
 - capacity, 121
 - delegation, 121, 122
 - guid, 122
 - health, 122
 - size, 122
 - used, 122
 - version, 122
- propriété delegation, désactivation, 222
- Propriété delegation, description, 121
- Propriété failmode, 122
- Propriété guid, description, 122
- Propriété health, description, 122
- Propriété size, description, 122
- Propriété used, description, 122
- Propriété version, description, 122
- Propriété ZFS
 - aclinherit, 144
 - aclmode, 144
 - atime, 144
 - available, 145
 - canmount, 145
 - Description détaillée, 153
 - checksum, 145
 - compression, 146
 - compressratio, 146
 - copies, 146
 - creation, 146
 - devices, 146
 - exec, 146
 - lecture seule, 150
 - mountpoint, 147
 - origin, 147
 - Pouvant être définie, 151
 - quota, 147
 - read-only, 147
 - recordsize, 148
 - Description détaillée, 153
 - referenced, 148
 - refquota, 148
 - refreservation, 148
 - reservation, 149
 - setuid, 149
 - sharenfs, 149
 - snapdir, 149
 - type, 149
 - used, 149
 - Description détaillée, 151
 - volblocksize, 150
 - volsize, 150
 - Description détaillée, 154
 - xattr, 150
 - zoned, 150
 - zoned, propriété
 - Description détaillée, 242
- Propriété ZFS en lecture seule
 - available, 145
 - compression, 146
 - creation, 146

Propriété ZFS en lecture seule (*Suite*)

- Description, 150
- origin, 147
- referenced, 148
- type, 149
- used, 149

Propriété ZFS pouvant être définie

- aclinherit, 144
- aclmode, 144
- atime, 144
- canmount, 145
 - Description détaillée, 153
- checksum, 145
- compression, 146
- copies, 146
- Description, 151
- devices, 146
- exec, 146
- mountpoint, 147
- quota, 147
- read-only, 147
- recordsize, 148
 - Description détaillée, 153
- refquota, 148
- refreservation, 148
- reservation, 149
- setuid, 149
- sharenfs, 149
- snappdir, 149
- used
 - Description détaillée, 151
- volblocksize, 150
- volsize, 150
 - Description détaillée, 154
- xattr, 150
- zoned, 150

Propriétés de ZFS

- Description, 143

Propriétés ZFS

- Définies par l'utilisateur
 - Description détaillée, 154
- Description des propriétés héritées, 143
- Gestion au sein d'une zone
 - Description, 241

Propriétés ZFS (*Suite*)

- Héritées, description, 143
- mounted, 147

Q

- quota, propriété, Description, 147
- Quota et réservation, Description, 170

R

- RAID-Z, Définition, 39
- read-only, propriété, Description, 147
- Réargenture, Définition, 39
- Réargenture et nettoyage des données,
 - Description, 251
- Réception
 - Données de système de fichiers ZFS (zfs receive)
 - (exemple), 186
- recordsize, propriété
 - Description, 148
 - Description détaillée, 153
- Récupération
 - Pool de stockage ZFS détruit
 - Exemple, 136
- referenced, propriété, Description, 148
- refquota, propriété, Description, 148
- refreservation, propriété, Description, 148
- Remplacement
 - Périphérique (zpool replace)
 - (exemple), 114, 266
 - Exemple, 262
 - Périphérique manquant
 - (exemple), 257
- Renommer
 - Instantané ZFS
 - Exemple, 179
- Réparation
 - Configuration ZFS endommagée
 - Description, 257
 - Dommages au niveau d'un pool
 - Description, 270

- Réparation (*Suite*)
 - Réparation d'un fichier ou répertoire endommagé
 - Description, 269
 - Système qui ne peut être initialisé
 - Description, 271
- reservation, propriété, Description, 149
- Résolution de problèmes
 - Corruption de données identifiée (`zpool status -v`)
 - Exemple, 255
 - Détermination du type de corruption de données (`zpool status -v`)
 - Exemple, 268
 - Détermination du type de panne de périphérique
 - Description, 259
 - Informations globales d'état des pools
 - Description, 253
 - Mode de panne ZFS, 247
 - Notification d'un périphérique reconnecté dans ZFS (`zpool online`)
 - Exemple, 258
 - Périphérique endommagé, 248
 - Rapport `syslog` de messages d'erreur ZFS, 256
 - Remplacement de périphérique (`zpool replace`)
 - Exemple, 262
 - Suppression des erreurs de périphérique (`zpool clear`)
 - Exemple, 260
- Restauration
 - ACL insignifiante sur un fichier ZFS (mode détaillé) (exemple), 205
 - Instantané ZFS
 - Exemple, 181
- S**
- Script
 - Sortie de pool de stockage ZFS
 - Exemple, 124
- Sémantique transactionnelle, Description, 35
- Séparation
 - Périphérique, d'un pool de stockage ZFS (`zpool detach`)
 - Exemple, 111
- setuid, propriété, Description, 149
- sharenfs, propriété
 - Description, 149, 169
- snapsdir, propriété, Description, 149
- Solaris Live Upgrade
 - Migration d'un système de fichiers racine, 67 (exemple), 70
 - Problèmes de migration d'un système de fichiers racine, 69
- Somme de contrôle, Définition, 37
- Stockage requis, Identification, 43
- Stockage sur pool, Description, 34
- Suppression
 - Erreurs de périphérique (`zpool clear`)
 - Exemple, 260
 - Suppression d'un droit, `zfs unallow`, 225
 - Suppression d'un périphérique
 - Pool de stockage ZFS
 - Exemple, 114
- Système de fichiers ZFS
 - Exemple, 142
- Système de fichiers, Définition, 38
- Système de fichiers racine ZFS, Problèmes de migration d'un système de fichiers racine, 69
- Système de fichiers ZFS
 - ACL dans un répertoire ZFS
 - Description détaillée, 200
 - Administration simplifiée
 - Description, 36
 - Ajout d'un système de fichiers ZFS à une zone non globale
 - Exemple, 239
 - Ajout d'un volume ZFS à une zone non globale (exemple), 240
 - Annulation du partage
 - exemple, 170
 - Clone
 - Création, 182
 - Définition, 37
 - Destruction, 183
 - Remplacement d'un système de fichiers (exemple), 183
 - Clones
 - Description, 182

Système de fichiers ZFS (*Suite*)

- Comptabilisation de l'espace d'un instantané, 181
- Configuration d'ACL dans un fichier ZFS (mode compact)
 - Exemple, 217
- Configuration d'héritage d'ACL dans un fichier ZFS (mode détaillé)
 - (exemple), 206
- Configuration requise pour l'installation et de Live Upgrade, 55
- Création
 - (exemple), 140
- Création d'un volume ZFS
 - (exemple), 235
- Création d'un volume ZFS en tant que périphérique de swap
 - (exemple), 236
- Définition d'ACL sur un fichier ZFS (mode compact)
 - Description, 216
- Définition d'un point de montage (zfs set mountpoint)
 - (exemple), 165
- Définition d'un point de montage hérité
 - (exemple), 166
- Définition d'une réservation
 - Exemple, 173
- Définition de la propriété quota
 - (exemple), 159
- Définition des ACL sur les fichiers ZFS
 - Description, 198
- Définition des ACL sur un fichier ZFS (mode détaillé)
 - Description, 200
- Définition d'atime, propriété
 - (exemple), 159
- Délégation d'un jeu de données à une zone non globale
 - (exemple), 240
- Démontage
 - (exemple), 168
- Description, 34, 139
- Destruction
 - (exemple), 141

Système de fichiers ZFS (*Suite*)

- Destruction avec les systèmes dépendants
 - (exemple), 142
- Données vérifiées par somme de contrôle
 - Description, 35
- Enregistrement d'un flux de données (zfs send)
 - (exemple), 185
- Envoi et réception
 - Description, 184
- Exigences d'attribution de noms de composant, 40
- Gestion de points de montage automatiques, 165
- Gestion de propriété au sein d'une zone
 - Description, 241
- Gestion des points de montage
 - Description, 164
- Gestion des points de montage hérités
 - Description, 164
- Héritage d'une propriété (zfs inherit)
 - (exemple), 159
- Initialisation d'un environnement d'initialisation ZFS avec boot -L et boot -Z
 - (exemple SPARC), 85
- Initialisation d'un système de fichiers racine
 - Description, 83
- Installation d'un système de fichiers racine, 54
- Installation initiale d'un système de fichiers racine ZFS, 57
- Installation JumpStart d'un système de fichiers racine, 63
- Instantané
 - Accès, 180
 - Création, 178
 - Définition, 39
 - Description, 177
 - Destruction, 179
 - Renommer, 179
 - Restauration, 181
- Jeu de données
 - Définition, 38
- Liste
 - (exemple), 156
- Liste de descendants
 - (exemple), 156

Système de fichiers ZFS (*Suite*)

- Liste de propriétés (`zfs list`)
 - (exemple), 160
- Liste de propriétés de script
 - (exemple), 163
- Liste de propriétés par valeur source
 - (exemple), 162
- Liste des types
 - Exemple, 158
- Liste sans en-tête
 - Exemple, 158
- Migration d'un système de fichiers racine avec Solaris Live Upgrade, 67
 - (exemple), 70
- Modification d'une ACL insignifiante dans un fichier ZFS (mode détaillé)
 - (Exemple), 202
- Modification du nom
 - Exemple, 142
- Montage
 - Exemple, 166
- Partage
 - Description, 169
 - Exemple, 169
- Périphériques de swap et de vidage
 - Ajustement de la taille, 82
 - Description, 81
 - Points à prendre en compte, 81
- Point de montage par défaut
 - (exemple), 140
- Profil de droits, 245
- Réception de flux de données (`zfs receive`)
 - (exemple), 186
- Restauration d'une ACL insignifiante sur un fichier ZFS (mode détaillé)
 - (exemple), 205
- Sémantique transactionnelle
 - Description, 35
- Somme de contrôle
 - Définition, 37
- Stockage sur pool
 - Description, 34
- Système de fichiers
 - Définition, 38

Système de fichiers ZFS (*Suite*)

- Type de jeu de données
 - Description, 157
- Utilisation sur un système Solaris doté de zones
 - Description, 239
- Volume
 - Définition, 40
- Système de fichiers ZFS (`zfs set quota`)
 - Définition d'un quota
 - Exemple, 171
- Système de stockage ZFS
 - Périphérique virtuel
 - Définition, 39
- Systèmes de fichiers ZFS
 - ACL sur un fichier ZFS
 - Description détaillée, 199

T

Terminologie

- Clone, 37
- Instantané, 39
- Jeu de données, 38
- Miroir, 38
- Périphérique virtuel, 39
- Pool, 38
- RAID-Z, 39
- Réargenture, 39
- Somme de contrôle, 37
- Système de fichiers, 38
- Volume, 40

Test

- Création de pool de stockage ZFS (`zpool create -n`)
 - Exemple, 104
- type, propriété, Description, 149
- Type de jeu de données, Description, 157

U

- used, propriété
 - Description, 149
 - Description détaillée, 151

Utilisateur, propriétés ZFS
Description détaillée, 154
Exemple, 154

V

volblocksize, propriété, Description, 150
volsize, propriété
Description, 150
Description détaillée, 154
Volume, Définition, 40
Volume ZFS
Description, 235
En tant que périphérique de swap, 236

X

xattr, propriété, Description, 150

Z

zfs allow
Affichage des droits délégués, 226
Description, 224
zfs create
(exemple), 46, 140
Description, 140
zfs destroy -r, (exemple), 142
zfs Destruction, (exemple), 141
zfs get, (exemple), 160
zfs get -H -o, (exemple), 163
zfs get -s, (exemple), 162
zfs inherit, (exemple), 159
zfs list
(exemple), 47, 156
zfs list -H, Exemple, 158
zfs list -r, (exemple), 156
zfs list -t, Exemple, 158
zfs mount, Exemple, 166
zfs promote, Promotion d'un clone (exemple), 183
zfs receive, (exemple), 186
zfs rename, Exemple, 142

zfs send, (exemple), 185
zfs set atime, (exemple), 159
zfs set compression, (exemple), 46
zfs set mountpoint
(exemple), 46, 165
zfs set mountpoint=legacy, (exemple), 166
zfs set quota
(exemple), 47
zfs set quota, (exemple), 159
zfs set quota
Exemple, 171
zfs set reservation, Exemple, 173
zfs set sharenfs, (exemple), 46
zfs set sharenfs=on, Exemple, 169
zfs unallow, Description, 225
zfs unmount, (exemple), 168

Zone

Ajout d'un système de fichiers ZFS à une zone non globale
Exemple, 239
Gestion de propriétés ZFS au sein d'une zone
Description, 241
Utilisation de systèmes de fichiers ZFS
Description, 239

zoned, propriété
Description, 150
Description détaillée, 242

Zones

Ajout d'un volume ZFS à une zone non globale
(exemple), 240
Délégation d'un jeu de données à une zone non globale
(exemple), 240
zoned, propriété
Description détaillée, 242
zpool add, Exemple, 106
zpool attach, Exemple, 109
zpool clear
Description, 114
Exemple, 114
zpool create
(exemple), 42, 44
Pool de base
(exemple), 97

zpool create (*Suite*)
 Pool de stockage mis en miroir
 (exemple), 97
 Pool de stockage RAID-Z
 Exemple, 98
zpool create -n
 Test
 Exemple, 104
zpool destroy, (exemple), 105
zpool detach, Exemple, 111
zpool export, (exemple), 131
zpool history, Exemple, 27
zpool import -a, (exemple), 132
zpool import -D, Exemple, 136
zpool import -d, (exemple), 134
zpool import *nom*, (exemple), 135
zpool iostat, pool complet, exemple, 125
zpool iostat -v, vdev, exemple, 126
zpool list
 (exemple), 44
 Description, 123
 Exemple, 124
zpool list -Ho name, Exemple, 124
zpool offline, (exemple), 112
zpool online, (exemple), 113
zpool replace, (exemple), 114
zpool status -v, (exemple), 129
zpool status -x, Exemple, 128
zpool upgrade, 137

