

Guide d'administration Oracle Solaris ZFS

Copyright © 2006, 2010, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Copyright © 2006, 2010, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Table des matières

Préface	11
1 Système de fichiers Oracle Solaris ZFS (introduction)	17
Nouveautés de ZFS	17
Scission d'un pool de stockage ZFS mis en miroir (<code>zpool split</code>)	18
Nouveau processus du système de fichiers ZFS	19
Modifications apportées à la commande <code>zpool list</code>	19
Récupération de pool de stockage ZFS	19
Améliorations des périphériques de journal ZFS	20
Configuration RAIDZ à triple parité (<code>raidz3</code>)	20
Conservation des instantanés ZFS	20
Améliorations du remplacement des périphériques ZFS	21
Prise en charge de l'installation de ZFS et Flash	23
Quotas d'utilisateurs et de groupes ZFS	23
Héritage direct ACL ZFS pour l'autorisation d'exécution	24
Améliorations apportées aux propriétés ZFS	24
Récupération de périphérique de journal ZFS	27
L'utilisation des périphériques de cache dans votre pool de stockage ZFS	28
Migration de zone dans un environnement ZFS	29
Prise en charge de l'installation et de l'initialisation de ZFS	29
Rétablissement d'un jeu de données sans démontage	30
Améliorations apportées à la commande <code>zfs send</code>	30
Quotas et réservations ZFS pour les données de système de fichiers uniquement	31
Propriétés de pool de stockage ZFS	31
Améliorations apportées à l'historique des commandes ZFS (<code>zpool history</code>)	32
Mise à niveau des systèmes de fichiers ZFS (<code>zfs upgrade</code>)	33
Administration déléguée de ZFS	34
Configuration de périphériques de journal ZFS distincts	34

Création de jeux de données ZFS intermédiaires	35
Amélioration de l'enfichage à chaud ZFS	36
Renommage récursif d'instantanés ZFS (<code>zfs rename -r</code>)	37
La compression <code>gzip</code> est disponible pour ZFS	38
Stockage de plusieurs copies de données utilisateur ZFS	38
Amélioration de la sortie de la commande <code>zpool status</code>	39
Améliorations de ZFS et Solaris iSCSI	39
Historique de commande ZFS (<code>zpool history</code>)	40
Améliorations de propriétés ZFS	41
Affichage de la totalité des informations de systèmes de fichiers ZFS	42
Nouvelle option <code>zfs receive -F</code>	42
Instantanés ZFS récursifs	42
Configuration RAID-Z à double parité (<code>raidz2</code>)	43
Disques hot spare pour périphériques de pool de stockage ZFS	43
Remplacement d'un système de fichiers ZFS par un clone ZFS (<code>zfs promote</code>)	43
Mise à niveau des pools de stockage ZFS (<code>zpool upgrade</code>)	43
Les commandes de sauvegarde et de restauration ZFS sont renommées.	44
Récupération de pools de stockage détruits	44
Intégration de ZFS au gestionnaire de pannes	44
Commande <code>zpool clear</code>	45
Format NFSv4 ACL compact	45
Outil de contrôle de système de fichiers (<code>fsstat</code>)	45
Gestion Web ZFS	46
Description de ZFS	47
Stockage ZFS mis en pool	47
Sémantique transactionnelle	47
Sommes de contrôle et données d'autorétablissement	48
Évolutivité inégalée	48
Instantanés ZFS	49
Administration simplifiée	49
Terminologie ZFS	49
Exigences d'attribution de noms de composants ZFS	52
2 Mise en route d'Oracle Solaris ZFS	53
Exigences et recommandations en matière de matériel et de logiciel ZFS	53

Création d'un système de fichiers ZFS basique	54
Création d'un pool de stockage ZFS	55
▼ Identification des exigences de stockage du pool de stockage ZFS	55
▼ Création d'un pool de stockage ZFS	56
Création d'une hiérarchie de systèmes de fichiers ZFS	56
▼ Détermination de la hiérarchie du système de fichiers ZFS	57
▼ Création de systèmes de fichiers ZFS	58
3 Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques	61
Granularité du système de fichiers ZFS	61
Comptabilisation de l'espace disque ZFS	62
Comportement d'espace saturé	62
Montage de système de fichiers ZFS	63
Gestion de volumes classique	63
Nouveau modèle ACL Solaris	63
4 Gestion des pools de stockage Oracle Solaris ZFS	65
Composants d'un pool de stockage ZFS	65
Utilisation de disques dans un pool de stockage ZFS	65
Utilisation de tranches dans un pool de stockage ZFS	67
Utilisation de fichiers dans un pool de stockage ZFS	68
Fonctions de réplication d'un pool de stockage ZFS	69
Configuration de pool de stockage mis en miroir	69
Configuration de pool de stockage RAID-Z	69
Pool de stockage ZFS hybride	71
Données d'autorétablissement dans une configuration redondante	71
Entrelacement dynamique dans un pool de stockage	71
Création et destruction de pools de stockage ZFS	72
Création d'un pool de stockage ZFS	72
Affichage des informations d'un périphérique virtuel de pool de stockage	77
Gestion d'erreurs de création de pools de stockage ZFS	78
Destruction de pools de stockage ZFS	81
Gestion de périphériques dans un pool de stockage ZFS	82
Ajout de périphériques à un pool de stockage	83
Connexion et séparation de périphériques dans un pool de stockage	88

Création d'un pool par scission d'un pool de stockage ZFS mis en miroir	89
Mise en ligne et mise hors ligne de périphériques dans un pool de stockage	93
Effacement des erreurs de périphérique de pool de stockage	95
Remplacement de périphériques dans un pool de stockage	96
Désignation des disques hot spare dans le pool de stockage	98
Gestion des propriétés de pool de stockage ZFS	104
Requête d'état de pool de stockage ZFS	107
Affichage des informations des pools de stockage ZFS	107
Visualisation des statistiques d'E/S des pools de stockage ZFS	111
Détermination de l'état de maintenance des pools de stockage ZFS	113
Migration de pools de stockage ZFS	116
Préparatifs de migration de pool de stockage ZFS	116
Exportation d'un pool de stockage ZFS	117
Définition des pools de stockage disponibles pour importation	117
Importation de pools de stockage ZFS à partir d'autres répertoires	119
Importation de pools de stockage ZFS	120
Récupération de pools de stockage ZFS détruits	121
Mise à niveau de pools de stockage ZFS	123
5 Installation et initialisation d'un système de fichiers racine Oracle Solaris ZFS	125
Installation et initialisation d'un système de fichiers racine Oracle Solaris ZFS (présentation)	126
Fonctions d'installation de ZFS	126
Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS	127
Installation d'un système de fichiers racine ZFS (installation initiale)	130
▼ Création d'un pool racine mis en miroir (post-installation)	136
Installation d'un système de fichiers racine ZFS (installation d'archive Oracle Solaris Flash) .	138
Installation d'un système de fichiers racine ZFS (installation d'Oracle Solaris JumpStart)	140
Mots-clés JumpStart pour ZFS	141
Exemples de profils JumpStart pour ZFS	143
Problèmes JumpStart pour ZFS	143
Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Oracle Solaris Live Upgrade)	144
Problèmes de migration ZFS avec Oracle Solaris Live Upgrade	145
Utilisation d'Oracle Solaris Live Upgrade pour migrer vers un système de fichiers racine	

ZFS (sans zones)	146
Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones (Solaris 10 10/08)	151
Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 5 10/09 ou supérieure)	156
Prise en charge ZFS des périphériques de swap et de vidage	167
Ajustement de la taille de vos périphériques de swap et de vidage ZFS	168
Dépannage du périphérique de vidage ZFS	170
Initialisation à partir d'un système de fichiers racine ZFS	171
Initialisation à partir d'un disque alternatif d'un pool racine ZFS mis en miroir	171
SPARC : Initialisation à partir d'un système de fichiers racine ZFS	172
x86 : initialisation à partir d'un système de fichiers ZFS racine	174
Résolution de problèmes de point de montage empêchant l'initialisation (Solaris 10 10/08)	175
Initialisation en vue d'une récupération d'un environnement racine ZFS	177
Restauration du pool racine ZFS ou des instantanés du pool racine	178
▼ Remplacement d'un disque dans le pool racine ZFS	178
▼ Création d'instantanés de pool racine	181
▼ Recréation d'un pool racine ZFS et restauration d'instantanés de pool racine	182
▼ Restauration des instantanés d'un pool racine à partir d'une initialisation de secours	184
6 Gestion des systèmes de fichiers Oracle Solaris ZFS	185
Gestion des systèmes de fichiers ZFS (présentation)	185
Création, destruction et renommage de systèmes de fichiers ZFS	186
Création d'un système de fichiers ZFS	186
Destruction d'un système de fichiers ZFS	187
Modification du nom d'un système de fichiers ZFS	188
Présentation des propriétés ZFS	189
Propriétés ZFS natives en lecture seule	198
Propriétés ZFS natives définies	199
Propriétés ZFS définies par l'utilisateur	202
Envoi de requêtes sur les informations des systèmes de fichiers ZFS	204
Affichage des informations de base des systèmes ZFS	204
Création de requêtes ZFS complexes	205
Gestion des propriétés ZFS	206
Définition des propriétés ZFS	206

Héritage des propriétés ZFS	207
Envoi de requêtes sur les propriétés ZFS	208
Montage et partage des systèmes de fichiers ZFS	211
Gestion des points de montage ZFS	211
Montage de système de fichiers ZFS	214
Utilisation de propriétés de montage temporaires	215
Démontage des systèmes de fichiers ZFS	215
Activation et annulation du partage des systèmes de fichiers ZFS	216
Définition des quotas et réservations ZFS	218
Définitions de quotas sur les systèmes de fichiers ZFS	219
Définition de réservations sur les systèmes de fichiers ZFS	222
7 Utilisation des instantanés et des clones ZFS Oracle Solaris	225
Présentation des instantanés ZFS	225
Création et destruction d'instantanés ZFS	226
Affichage et accès des instantanés ZFS	229
Restauration d'un instantané ZFS	231
Présentation des clones ZFS	232
Création d'un clone ZFS	232
Destruction d'un clone ZFS	233
Remplacement d'un système de fichiers ZFS par un clone ZFS	233
Envoi et réception de données ZFS	234
Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde	235
Envoi d'un instantané ZFS	235
Réception d'un instantané ZFS	236
Envoi et réception de flux d'instantanés ZFS complexes	237
8 Utilisation des ACL pour protéger les fichiers Oracle Solaris ZFS	241
Nouveau modèle ACL Solaris	241
Descriptions de syntaxe pour la configuration des ACL	243
Héritage d'ACL	246
Propriétés ACL	247
Configuration d'ACL dans des fichiers ZFS	248
Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé	250
Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé	255

Configuration et affichage d'ACL dans des fichiers ZFS en format compact	262
9 Administration déléguée de ZFS	269
Présentation de l'administration déléguée de ZFS	269
Désactivation des droits délégués de ZFS	270
Délégation de droits ZFS	270
Délégation des autorisations ZFS (<code>zfs allow</code>)	272
Suppression des droits délégués de ZFS (<code>zfs unallow</code>)	273
Utilisation de l'administration déléguée de ZFS	274
Délégation de droits ZFS (exemples)	274
Affichage des droits ZFS délégués (exemples)	278
Suppression de droits ZFS (exemples)	279
10 Rubriques avancées Oracle Solaris ZFS	281
Volumes ZFS	281
Utilisation d'un volume ZFS en tant que périphérique de swap ou de dump	282
Utilisation d'un volume ZFS en tant que cible iSCSI Solaris	283
Utilisation de ZFS dans un système Solaris avec zones installées	284
Ajout de systèmes de fichiers ZFS à une zone non globale	285
Délégation de jeux de données à une zone non globale	286
Ajout de volumes ZFS à une zone non globale	286
Utilisation de pools de stockage ZFS au sein d'une zone	287
Gestion de propriétés ZFS au sein d'une zone	287
Explication de la propriété <code>zoned</code>	288
Utilisation de pools racine ZFS de remplacement	290
Création de pools racine de remplacement ZFS	290
Importation de pools racine de remplacement	291
Profils de droits ZFS	291
11 Dépannage d'Oracle Solaris ZFS et récupération de pool	293
Identification des défaillances ZFS	293
Périphériques manquants dans un pool de stockage ZFS	294
Périphériques endommagés dans un pool de stockage ZFS	294
Données ZFS corrompues	294

Contrôle de l'intégrité d'un système de fichiers ZFS	295
Réparation du système de fichiers	295
Validation du système de fichiers	295
Contrôle du nettoyage de données ZFS	296
Résolution de problèmes avec le système de fichiers ZFS	297
Recherche de problèmes éventuels dans un pool de stockage ZFS	299
Consultation de la sortie de <code>zpool status</code>	299
Rapport système de messages d'erreur ZFS	302
Réparation d'une configuration ZFS endommagée	302
Réparation d'un périphérique manquant	303
Reconnexion physique d'un périphérique	304
Notification relative à la disponibilité de périphériques dans ZFS	304
Remplacement ou réparation d'un périphérique endommagé	305
Détermination du type de panne de périphérique	305
Suppression des erreurs transitoires	307
Remplacement d'un périphérique dans un pool de stockage ZFS	307
Réparation de données endommagées	314
Identification du type de corruption de données	315
Réparation d'un fichier ou répertoire corrompu	316
Réparation de dommages présents dans l'ensemble du pool de stockage ZFS	317
Réparation d'un système impossible à réinitialiser	319
A Descriptions des versions d'Oracle Solaris ZFS	321
Présentation des versions ZFS	321
Versions de pool ZFS	321
Versions du système de fichiers ZFS	322
Index	325

Préface

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

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

Remarque – Cette version d'Oracle Solaris prend en charge les systèmes utilisant les architectures de processeur SPARC et x86 : 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 Oracle Solaris. Il est recommandé de savoir utiliser le système d'exploitation Oracle Solaris ou toute autre version UNIX.

Organisation de ce document

Le tableau suivant décrit les chapitres de ce document.

Chapitre	Description
Chapitre 1, “Système de fichiers Oracle Solaris ZFS (introduction)”	Présente le système de fichiers ZFS, ses fonctionnalités et ses avantages. Il aborde également des concepts de base, ainsi que la terminologie.
Chapitre 2, “Mise en route d’Oracle Solaris ZFS”	Décrit étape par étape les instructions d’une configuration ZFS de base contenant des pools et des systèmes de fichiers simples. Ce chapitre indique également le matériel et logiciels requis pour la création de systèmes de fichiers ZFS.
Chapitre 3, “Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques”	Identifie les fonctionnalités importantes qui différencient le système de fichiers ZFS des systèmes de fichiers standard. Lors de l’utilisation d’outils classiques avec le système de fichiers ZFS, la compréhension de ces différences clés permet d’éviter les confusions.
Chapitre 4, “Gestion des pools de stockage Oracle Solaris ZFS”	Décrit en détail les méthodes de création et d’administration de pools de stockage ZFS.
Chapitre 5, “Installation et initialisation d’un système de fichiers racine Oracle Solaris 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 racine ZFS à l’aide de Solaris Live Upgrade est également abordée.
Chapitre 6, “Gestion des systèmes de fichiers Oracle Solaris ZFS”	Décrit en détail les méthodes de gestion de systèmes de fichiers ZFS. Ce chapitre décrit des concepts tels que la disposition hiérarchique de systèmes de fichiers, l’héritage de propriétés, la gestion automatique de points de montage et les interactions de partage.
Chapitre 7, “Utilisation des instantanés et des clones ZFS Oracle Solaris”	Décrit les méthodes de création et d’administration d’instantanés ZFS et de clones.
Chapitre 8, “Utilisation des ACL pour protéger les fichiers Oracle Solaris ZFS”	Explique comment utiliser des listes de contrôle d’accès (ACL, Access Control List) pour la protection des fichiers ZFS en fournissant davantage de droits d’accès granulaires que les droits d’accès UNIX standard.
Chapitre 9, “Administration déléguée de ZFS”	Explique comment utiliser les fonctions de l’administration déléguée de ZFS pour permettre aux utilisateurs ne disposant pas des autorisations nécessaires d’effectuer des tâches d’administration ZFS.
Chapitre 10, “Rubriques avancées Oracle Solaris ZFS”	Explique comment utiliser des volumes et des systèmes de fichiers ZFS dans un système Oracle Solaris comportant des zones et comment utiliser les pools racine de remplacement.

Chapitre	Description
Chapitre 11, “Dépannage d'Oracle Solaris ZFS et récupération de pool”	Explique comment identifier des modes de défaillance de ZFS et les solutions existantes. Les étapes de prévention de ces défaillances sont également abordées.
Annexe A, “Descriptions des versions d'Oracle Solaris ZFS”	Décrit les versions ZFS disponibles, les fonctionnalités de chacune d'entre elles et le SE Solaris fournissant la version et les fonctionnalités ZFS.

Documentation connexe

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

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

Documentation, support et formation

Pour obtenir des ressources supplémentaires, reportez-vous aux sites Web suivants :

- Documentation (<http://docs.sun.com>)
- Assistance (<http://www.oracle.com/us/support/systems/index.html>)
- Formation (<http://education.oracle.com>) - Cliquez sur le lien Sun dans la barre de navigation à gauche.

Oracle vous encourage à envoyer vos commentaires.

Oracle vous encourage à envoyer vos commentaires et à apporter des suggestions sur la qualité et l'utilité de sa documentation. Si vous constatez des erreurs ou que vous avez d'autres suggestions d'amélioration, rendez-vous à l'adresse <http://docs.sun.com> et cliquez le lien Feedback. Indiquez le titre et le numéro de référence de la documentation en même temps que le chapitre, une section et numéro de page, le cas échéant. Veuillez nous indiquer si vous souhaitez recevoir une réponse.

Le site [réseau technologie d'Oracle](http://www.oracle.com/technetwork/index.html) (<http://www.oracle.com/technetwork/index.html>) propose un éventail de ressources liées aux logiciels Oracle :

- Discutez de problèmes techniques et trouvez des solutions sur les [forums de discussion](http://forums.oracle.com) (<http://forums.oracle.com>).
- Passez à la pratique grâce aux didacticiels étape par étape avec [Oracle By Example](http://www.oracle.com/technology/obe/start/index.html) (<http://www.oracle.com/technology/obe/start/index.html>).
- Téléchargez [des échantillons de code](http://www.oracle.com/technology/sample_code/index.html) (http://www.oracle.com/technology/sample_code/index.html).

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% Vous avez reçu du courrier.</code>
AaBbCc123	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	<code>nom_machine% su</code> 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 <i>nom_fichier</i></code> .
<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 l'invite système UNIX par défaut et l'invite superutilisateur pour les shells inclus dans le SE Oracle Solaris. L'invite système par défaut s'affichant dans les exemples de commandes dépend de la version Oracle Solaris.

TABLEAU P-2 Invites de shell

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

Système de fichiers Oracle Solaris ZFS (introduction)

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

Il contient les sections suivantes :

- “Nouveautés de ZFS” à la page 17
- “Description de ZFS” à la page 47
- “Terminologie ZFS” à la page 49
- “Exigences d’attribution de noms de composants ZFS” à la page 52

Nouveautés de ZFS

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

- “Scission d’un pool de stockage ZFS mis en miroir (`zpool split`)” à la page 18
- “Nouveau processus du système de fichiers ZFS” à la page 19
- “Modifications apportées à la commande `zpool list`” à la page 19
- “Récupération de pool de stockage ZFS” à la page 19
- “Améliorations des périphériques de journal ZFS” à la page 20
- “Configuration RAIDZ à triple parité (`raidz3`)” à la page 20
- “Conservation des instantanés ZFS” à la page 20
- “Améliorations du remplacement des périphériques ZFS” à la page 21
- “Prise en charge de l’installation de ZFS et Flash” à la page 23
- “Quotas d’utilisateurs et de groupes ZFS” à la page 23
- “Héritage direct ACL ZFS pour l’autorisation d’exécution” à la page 24
- “Améliorations apportées aux propriétés ZFS” à la page 24
- “Récupération de périphérique de journal ZFS” à la page 27
- “L’utilisation des périphériques de cache dans votre pool de stockage ZFS” à la page 28
- “Migration de zone dans un environnement ZFS” à la page 29
- “Prise en charge de l’installation et de l’initialisation de ZFS” à la page 29
- “Rétablissement d’un jeu de données sans démontage” à la page 30

- “Améliorations apportées à la commande `zfs send`” à la page 30
- “Quotas et réservations ZFS pour les données de système de fichiers uniquement” à la page 31
- “Propriétés de pool de stockage ZFS” à la page 31
- “Améliorations apportées à l'historique des commandes ZFS (`zpool history`)” à la page 32
- “Mise à niveau des systèmes de fichiers ZFS (`zfs upgrade`)” à la page 33
- “Administration déléguée de ZFS” à la page 34
- “Configuration de périphériques de journal ZFS distincts” à la page 34
- “Création de jeux de données ZFS intermédiaires” à la page 35
- “Amélioration de l'enfichage à chaud ZFS” à la page 36
- “Renommage récursif d'instantanés ZFS (`zfs rename -r`)” à la page 37
- “La compression `gzip` est disponible pour ZFS” à la page 38
- “Stockage de plusieurs copies de données utilisateur ZFS” à la page 38
- “Amélioration de la sortie de la commande `zpool status`” à la page 39
- “Améliorations de ZFS et Solaris iSCSI” à la page 39
- “Historique de commande ZFS (`zpool history`)” à la page 40
- “Améliorations de propriétés ZFS” à la page 41
- “Affichage de la totalité des informations de systèmes de fichiers ZFS” à la page 42
- “Nouvelle option `zfs receive -F`” à la page 42
- “Instantanés ZFS récursifs” à la page 42
- “Configuration RAID-Z à double parité (`raidz2`)” à la page 43
- “Disques hot spare pour périphériques de pool de stockage ZFS” à la page 43
- “Remplacement d'un système de fichiers ZFS par un clone ZFS (`zfs promote`)” à la page 43
- “Mise à niveau des pools de stockage ZFS (`zpool upgrade`)” à la page 43
- “Les commandes de sauvegarde et de restauration ZFS sont renommées.” à la page 44
- “Récupération de pools de stockage détruits” à la page 44
- “Intégration de ZFS au gestionnaire de pannes” à la page 44
- “Commande `zpool clear`” à la page 45
- “Format NFSv4 ACL compact” à la page 45
- “Outil de contrôle de système de fichiers (`fsstat`)” à la page 45
- “Gestion Web ZFS” à la page 46

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

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

Pour plus d'informations, reportez-vous à la section “Création d'un pool par scission d'un pool de stockage ZFS mis en miroir” à la page 89.

Nouveau processus du système de fichiers ZFS

Oracle Solaris 10 9/10 : dans cette version de Solaris, chaque pool de stockage ZFS est associé à un processus `zpool -poolname`. Les threads de ce processus correspondent à des threads de traitement d'E/S du pool permettant de gérer des tâches d'E/S associées au pool comme la compression et les sommes de contrôle. Le but de ce processus est d'indiquer l'utilisation de la CPU de chaque pool de stockage. Pour plus d'informations sur ces processus, reportez-vous aux commandes `ps` et `prstat`. Ces processus sont uniquement disponibles dans la zone globale. Pour de plus amples informations, reportez-vous à la section [SDC\(7\)](#).

Modifications apportées à la commande `zpool list`

Oracle Solaris 10 9/10 : dans cette version de Solaris, la sortie de la commande `zpool list` a été modifiée pour obtenir des informations plus précises concernant l'allocation d'espace.

Exemple :

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

Les champs `USED` et `AVAIL` précédents ont été remplacés par les champs `ALLOC` et `FREE`.

Le champ `ALLOC` permet d'identifier la quantité d'espace physique allouée à tous les jeux de données et aux métadonnées internes. Le champ `FREE` permet d'identifier la quantité d'espace inutilisé dans le pool.

Pour plus d'informations, reportez-vous à la section “[Affichage des informations des pools de stockage ZFS](#)” à la page 107.

Récupération de pool de stockage ZFS

Oracle Solaris 10 9/10 : un pool de stockage peut être endommagé si les périphériques sous-jacents sont indisponibles, si une panne de courant survient ou si un nombre de périphériques plus important que les périphériques pris en charge échoue dans une configuration ZFS redondante. Cette version présente de nouvelles fonctionnalités de commande permettant de récupérer votre pool de stockage endommagé. Cependant, l'utilisation de ces fonctions de récupération signifie que les dernières transactions restantes survenues lors de l'interruption de service du pool peuvent être perdues.

Les deux commandes `zpool clear` et `zpool import` prennent en charge l'option `-F` pour récupérer éventuellement un pool endommagé. En outre, l'exécution de la commande `zpool status`, `zpool clear` ou `zpool import` signale automatiquement un pool endommagé et ces commandes expliquent comment récupérer le pool.

Pour plus d'informations, reportez-vous à la section “[Réparation de dommages présents dans l'ensemble du pool de stockage ZFS](#)” à la page 317.

Améliorations des périphériques de journal ZFS

Oracle Solaris 10 9/10 : les améliorations suivantes ont été apportées aux périphériques de journal :

- Propriété `logbias` – Cette propriété peut être utilisée pour fournir un conseil au système de fichiers ZFS concernant la gestion des demandes synchrones d'un jeu de données spécifique. Si la propriété `logbias` est définie sur `latency`, ZFS utilise des périphériques de journal distincts dans le pool pour gérer les demandes à faible latence. Si la propriété `logbias` est définie sur `throughput`, le système de fichiers ZFS n'utilise pas de périphériques de journal distincts dans le pool. Au lieu de cela, le système de fichiers ZFS optimise les opérations synchrones pour traiter globalement le pool et utiliser efficacement les ressources. La valeur par défaut est `latency`. Nous vous recommandons d'utiliser la valeur par défaut pour la plupart des configurations. La valeur `logbias=throughput` peut éventuellement améliorer les performances d'enregistrement des fichiers de la base de données.
- Suppression d'un périphérique de journal : vous pouvez maintenant supprimer un périphérique de journal à partir d'un pool de stockage ZFS à l'aide de la commande `zpool remove`. Vous pouvez supprimer un périphérique de journal unique en spécifiant le nom de ce dernier. Vous pouvez supprimer un périphérique de journal mis en miroir en spécifiant le miroir de niveau supérieur du journal. Lorsqu'un périphérique de journal distinct est supprimé du système, les enregistrements de transaction ZIL sont enregistrés dans le pool principal.

Les périphériques virtuels de niveau supérieur redondants sont maintenant identifiés par un identificateur numérique. Par exemple, dans un pool de stockage mis en miroir composé de deux disques, le périphérique virtuel de niveau supérieur porte le nom `mirror-0`.

Pour plus d'informations, reportez-vous à l'[Exemple 4-3](#).

Configuration RAIDZ à triple parité (`raidz3`)

Oracle Solaris 10 9/10 : dans cette version de Solaris, une configuration RAID-Z redondante peut maintenant disposer d'une parité simple, double, ou triple. En d'autres termes, le système peut supporter une, deux ou trois pannes de périphérique sans perte de données. Le mot-clé `raidz3` permet de spécifier une configuration RAID-Z à parité triple. Pour plus d'informations, reportez-vous à la section "[Création d'un pool de stockage RAID-Z](#)" à la page 74.

Conservation des instantanés ZFS

Oracle Solaris 10 9/10 : si vous implémentez plusieurs stratégies d'instantanés automatiques et que les instantanés les plus anciens sont détruits par inadvertance par la commande `zfs receive` car ils n'existent plus du côté de l'envoi, vous pouvez utiliser la fonction de conservation des instantanés dans cette version de Solaris.

La conservation d'un instantané empêche sa destruction. En outre, cette fonction permet de supprimer un instantané contenant des clones en attendant la suppression du dernier clone à l'aide de la commande `zfs destroy -d`.

Vous pouvez conserver un instantané ou un jeu d'instantanés. Par exemple, la syntaxe suivante insère une balise de conservation `keep` sur `citerne/home/cindys/snap@1`.

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

Pour plus d'informations, reportez-vous à la section [“Conservation des clichés ZFS”](#) à la page 227.

Améliorations du remplacement des périphériques ZFS

Oracle Solaris 10 9/10 : dans cette version de Solaris, un événement système ou *sysevent* est fourni lorsqu'un périphérique sous-jacent est étendu. Le système de fichiers ZFS a été amélioré pour reconnaître des événements et ajuste le pool en fonction de la nouvelle taille du LUN (Logical Unit Number) étendu, selon la définition de la propriété `autoexpand`. Vous pouvez utiliser la propriété `autoexpand` du pool pour activer ou désactiver l'extension automatique du pool lorsqu'un événement d'extension LUN dynamique est reçu.

Ces fonctions permettent d'étendre un LUN et le pool correspondant peut accéder à l'espace étendu sans avoir à exporter et importer le pool ou réinitialiser le système.

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

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

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

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

La propriété `autoexpand` est désactivée par défaut. Vous pouvez décider d'étendre ou non le LUN.

Vous pouvez également étendre un LUN grâce à la commande `zpool online -e`. Exemple :

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

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

```

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

```

Dans l'exemple ci-dessus, vous pouvez également étendre le LUN sans activer la propriété `autoexpand`. Pour ce faire, vous devez utiliser la commande `zpool online -e`, même si le périphérique est déjà en ligne. Exemple :

```

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

```

Cette version comprend les améliorations suivantes concernant le remplacement de périphérique :

- Dans les versions précédentes, le système de fichiers ZFS ne pouvait pas remplacer un disque existant par un autre disque ou connecter un disque, si la taille du disque de remplacement était légèrement différente. Dans cette version, vous pouvez remplacer un disque existant par un autre disque ou connecter un nouveau disque qui dispose théoriquement de la même taille, à condition que le pool ne soit pas déjà plein.
- Dans cette version, vous ne devez pas réinitialiser le système ou exporter/importer un pool pour étendre un LUN. Comme décrit ci-dessus, vous pouvez activer la propriété `autoexpand` ou utiliser la commande `zpool online -e` pour étendre le LUN à sa taille maximale.

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 96.

Prise en charge de l'installation de ZFS et Flash

Version Solaris 10 10/09 : dans cette version de Solaris, vous pouvez définir un profil JumpStart pour identifier une archive Flash d'un pool racine ZFS. Pour plus d'informations, reportez-vous à la section [“Installation d'un système de fichiers racine ZFS \(installation d'archive Oracle Solaris Flash\)”](#) à la page 138.

Quotas d'utilisateurs et de groupes ZFS

Version Solaris 10 10/09 : dans les versions précédentes de Solaris, vous pouviez appliquer des quotas et des réservations aux systèmes de fichiers ZFS pour gérer et réserver de l'espace disque.

Dans cette version de Solaris, vous pouvez définir un quota sur la quantité d'espace disque utilisée par les fichiers appartenant à un utilisateur ou à un groupe spécifique. Vous pouvez envisager la définition de quotas d'utilisateurs et de groupes dans un environnement contenant un grand nombre d'utilisateurs ou de groupes.

Vous pouvez définir un quota d'utilisateurs en utilisant la propriété `zfs userquota`. Pour définir le quota d'un groupe, utilisez la propriété `zfs groupquota`. Exemple :

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

Vous pouvez afficher le paramètre de quota actuel d'un utilisateur ou d'un groupe comme suit :

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

Affichez des informations générales sur les quotas comme suit :

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

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

Vous pouvez afficher l'utilisation de l'espace disque d'un utilisateur en affichant la propriété `userused@user`. Vous pouvez afficher l'utilisation de l'espace disque d'un groupe avec la propriété `groupused@group`. Exemple :

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

Pour plus d'informations sur la définition des quotas d'utilisateurs, consultez la section [“Définition des quotas et réservations ZFS”](#) à la page 218.

Héritage direct ACL ZFS pour l'autorisation d'exécution

Versión Solaris 10 10/09 : dans les versions précédentes de Solaris, vous pouviez appliquer l'héritage ACL afin que tous les fichiers soient créés avec l'autorisation 0664 ou 0666. Dans cette version, si vous souhaitez éventuellement ajouter le bit d'exécution lors de la création du fichier à l'ACL héritée, vous pouvez définir le mode `aclinherit` pour qu'il transmette l'autorisation d'exécution à l'ACL héritée.

Si `aclinherit=passthrough-x` est activée sur un jeu de données ZFS, vous pouvez inclure une autorisation d'exécution sur un fichier de sortie qui est générée par les outils de compilation `cc` ou `gcc`. Si l'ACL héritée ne comprend pas les autorisations d'exécution, la sortie exécutable du compilateur ne sera pas exécutable jusqu'à ce que vous utilisiez la commande `chmod` pour modifier les autorisations du fichier.

Pour plus d'informations, reportez-vous à l'[Exemple 8-12](#).

Améliorations apportées aux propriétés ZFS

Versions Solaris 10 10/09 et Oracle Solaris 10 9/10 : les améliorations apportées aux systèmes de fichiers ZFS sont incluses dans ces versions.

- **Améliorations des propriétés de flux d'instantané ZFS** - Vous pouvez définir une propriété reçue différente de la définition de la propriété locale. Par exemple, vous pouvez recevoir un flux avec la propriété de compression désactivée, mais vous souhaitez que cette propriété soit activée dans le système de fichiers destinataire. Cela signifie que la valeur de compression du flux reçue est définie sur `off` et que la valeur de compression locale est définie sur `on`. Étant donné que la valeur locale remplace la valeur reçue, vous ne devez pas définir les paramètres du côté de l'envoi pour qu'ils remplacent la valeur du côté de la réception. La commande `zfs get` affiche la valeur effective de la propriété de compression sous la colonne `VALUE`.

Les nouvelles options et propriétés des commandes ZFS prenant en charge les valeurs des propriétés locales et des propriétés d'envoi sont les suivantes :

- Utilisez la commande `zfs inherit -S` pour rétablir la valeurs de propriété locale reçue, le cas échéant. Si une propriété ne reçoit aucune valeur, le comportement de la commande `zfs inherit -S` est le même que la commande `zfs inherit` sans l'option `-S`. Si la propriété ne reçoit aucune valeur, la commande `zfs inherit` masque la valeur reçue par la valeur héritée jusqu'à ce que l'émission d'une commande `zfs inherit -S` rétablisse la valeur reçue.
- Vous pouvez utiliser la commande `zfs get -o` pour prendre en compte la nouvelle colonne RECEIVED ajoutée. Vous pouvez également utiliser la commande `zfs get -o all` pour ajouter toutes les colonnes, y compris la colonne RECEIVED.
- Vous pouvez utiliser l'option `zfs send -p` pour ajouter des propriétés dans le flux d'envoi sans l'option `-R`.

En outre, vous pouvez utiliser l'option `zfs send -e` pour utiliser le dernier élément du nom de l'instantané envoyé pour déterminer le nom du nouvel instantané. L'exemple suivant envoie l'instantané `poola/bee/cee@1` au système de fichiers `poold/eee` et utilise uniquement le dernier élément (`cee@1`) du nom de l'instantané pour créer le système de fichiers et l'instantané reçus.

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

- **Définition des propriétés de système de fichiers ZFS lors de la création du pool** : vous pouvez définir les propriétés d'un système de fichiers ZFS lors de la création du pool de stockage. Dans l'exemple suivant, la compression est activée sur le système de fichiers ZFS qui est créé lors de la création du pool:


```
# zpool create -O compression=on pool mirror c0t1d0 c0t2d0
```
- **Définition des propriétés de cache sur un système de fichiers ZFS** : deux nouvelles propriétés de système de fichiers ZFS vous permettent de déterminer quels sont les éléments mis en mémoire cache dans le cache principal (ARC) ou le cache secondaire (L2ARC). Les propriétés du cache sont définies comme suit :
 - `Primarycache` : contrôle les éléments qui sont mis en cache dans le cache principal.
 - `Secondarycache` : contrôle les éléments qui sont mis en cache dans le cache secondaire.

- Les valeurs possibles de ces deux propriétés sont `all`, `none` et `metadata`. Si elles sont définies sur `all`, les données d'utilisateur et les métadonnées sont mises en cache. Si elle est définie sur `none`, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si elles sont définies sur `metadata`, seules les métadonnées sont mises en cache. La valeur par défaut est `all`.

Vous pouvez définir ces propriétés sur un système de fichiers existant ou lors de la création d'un système de fichiers. Exemple :

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

Lorsque ces propriétés sont définies sur des systèmes de fichiers existants, seule la nouvelle E/S est mise en cache en fonction de la valeur de ces propriétés.

Certains environnements de bases de données pourraient bénéficier de la non-mise en cache des données d'utilisateur. Vous devrez déterminer si la définition des propriétés du cache est appropriée pour votre environnement.

- **Affichage des propriétés de comptabilisation d'espace** : de nouvelles propriétés de système de fichiers en lecture seule vous aident à identifier l'utilisation de l'espace disque pour les clones, les systèmes de fichiers et les volumes, mais pas les instantanés. Les propriétés sont les suivantes :
 - `usedbychildren` : affiche l'espace disque occupé par les niveaux enfant de ce jeu de données. Ce dernier peut être libéré si tous les niveaux enfants du jeu de données sont détruits. L'abréviation de la propriété est `usedchild`.
 - `usedbydataset` : affiche l'espace disque occupé par le jeu de données, lors de la première destruction d'un instantané ou de la suppression d'une propriété `refreservation`. Cet espace peut être libéré si le jeu de données est détruit. L'abréviation de la propriété est `usedds`.
 - `usedbyrefreservation` : affiche l'espace disque occupé par le jeu `refreservation` de ce jeu de données. Cet espace peut être libéré si le jeu `refreservation` est supprimé. L'abréviation de la propriété est `usedrefreserv`.
 - `usedbysnapshots` : affiche l'espace disque occupé par les instantanés de ce jeu de données. Cet espace peut être libéré si l'ensemble des instantanés de ce jeu de données est détruit. Notez qu'il ne s'agit pas de la somme des propriétés `used` des instantanés, car l'espace disque peut être partagé par plusieurs instantanés. L'abréviation de la propriété est `usedsnap`.

Ces nouvelles propriétés divisent la valeur de la propriété `used` dans les divers éléments qui utilisent de l'espace disque. En particulier, la valeur de la propriété `used` est divisée comme suit :

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

Vous pouvez visualiser ces propriétés à l'aide de la commande `zfs list -o space`.

Exemple :

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

La commande précédente est équivalente à la commande `zfs list`

- o name,avail,used,usedsnap,useddes,usedrefreserv,usedchild -t filesystem,volume.

- **Liste d'instantanés**– La propriété de `pool listsnapshots` vérifie si les informations sur les instantanés sont affichées par la commande `zfs list`. La valeur par défaut est `on`, ce qui signifie que les informations sur les instantanés sont affichées par défaut.

Si votre système contient un nombre important d'instantanés ZFS et que vous souhaitez désactiver l'affichage des informations concernant les instantanés dans la commande `zfs list`, désactivez la propriété `listsnapshots` comme suit :

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

Si vous désactivez la propriété `listsnapshots`, vous pouvez utiliser la commande `zfs list -t snapshots` pour afficher les informations sur l'instantané. Exemple :

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

Récupération de périphérique de journal ZFS

Versión Solaris 10 10/09 : dans cette version, ZFS identifie les défaillances de journal d'intention dans la sortie de la commande `zpool status`. Le composant FMA (Fault Management Architecture) signale également ces erreurs. ZFS et FMA décrivent comment récupérer les données en cas de défaillance du journal d'intention.

Par exemple, si le système s'arrête soudainement avant que les opérations d'écriture synchrone ne soient affectées à un pool disposant d'un périphérique de journal distinct, un message tel que le suivant s'affiche :

```
# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
       Waiting for administrator intervention to fix the faulted pool.
```

action: Either restore the affected device(s) and run 'zpool online',
or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:

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

Vous pouvez résoudre la panne du périphérique de journal de l'une des façons suivantes :

- Remplacez ou récupérez le périphérique de journal. Dans cet exemple, le périphérique de journal est c0t5d0.
- Mettez le périphérique de journal en ligne.


```
# zpool online pool c0t5d0
```
- Réinitialisez la condition d'erreur de périphérique de journal défaillante.


```
# zpool clear pool
```

Pour effectuer une récupération suite à cette erreur sans remplacer le périphérique de journal défaillant, vous pouvez effacer l'erreur à l'aide de la commande `zpool clear`. Dans ce scénario, le pool fonctionnera en mode dégradé et les enregistrements de journal seront enregistrés dans le pool principal jusqu'à ce que le périphérique de journal distinct soit remplacé.

Envisagez d'utiliser des périphériques de journal mis en miroir afin d'éviter un scénario de défaillance de périphérique de journal.

L'utilisation des périphériques de cache dans votre pool de stockage ZFS

Version Solaris 10 10/09 : dans cette version de Solaris, vous pouvez créer un pool et spécifier *les périphériques de cache*, utilisés pour mettre en cache les données du pool de stockage.

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

Vous pouvez spécifier un ou plusieurs périphériques de cache à la création du pool. Exemple :

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

```
scrub: none requested
config:
```

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

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

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

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

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

Pour plus d'informations, reportez-vous à la section [“Création d'un pool de stockage ZFS avec des périphériques de cache”](#) à la page 76 et à l'[Exemple 4-4](#).

Migration de zone dans un environnement ZFS

Version Solaris 10 5/09 : cette version étend la prise en charge des zones de migration dans un environnement ZFS avec Oracle Solaris Live Upgrade. Pour plus d'informations, reportez-vous à la section [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones \(version Solaris 5 10/09 ou supérieure\)”](#) à la page 156.

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

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

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 également utiliser la fonction Oracle Solaris 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 [Chapitre 5, “Installation et initialisation d'un système de fichiers racine Oracle Solaris ZFS”](#).

Pour obtenir une liste des problèmes connus avec cette version, consultez le site Web à l'adresse suivante :

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

Reportez-vous également aux notes de version de Solaris 10 10/08.

Rétablissement d'un jeu de données sans démontage

Solaris 10 10/08 : cette version permet d'annuler (roll back) un jeu de données sans devoir au préalable le démonter. L'option `zfs rollback -f` n'est ainsi plus nécessaire pour forcer l'opération de démontage. 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 : les améliorations suivantes ont été apportées à la commande `zfs send` dans cette version. Grâce à cette commande, vous pouvez désormais effectuer les tâches suivantes :

- 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
```

Cette syntaxe comprend tous les instantanés incrémentiels de `fs@snapA` à `fs@snapD`, qui 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/fscloonesnap-I
.
.
# zfs receive -F pool/clone < /snaps/fscloonesnap-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. Exemple :

```
# 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 237.

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 disque.

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

Par exemple, en définissant la limite de la propriété `refquota` de `studentA` sur 10 Go, vous spécifiez une limite fixe d'espace disque *référéncé* de 10 Go. 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 "[Définition des quotas et réservations ZFS](#)" à la page 218.

Propriétés de pool de stockage ZFS

Version Solaris 10 10/08 : les propriétés de pool de stockage ZFS ont été introduites dans une version antérieure. Cette version contient deux propriétés : `cacheFile` et `failmode`.

La section suivante décrit les propriétés du nouveau pool de stockage de cette version :

- Propriété `cacheFile` - Cette propriété vérifie l'emplacement du cache des informations de configuration du 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 `cache file` 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` : cette propriété détermine 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, reportez-vous au [Tableau 4-1](#).

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 des événements du système de fichiers ZFS s'affichent. Exemple :

```
# zpool history
History for 'rpool':
2010-06-23.09:30:12 zpool create -f -o failmode=continue -R /a -m legacy -o
cache file=/tmp/root/etc/zfs/zpool.cache rpool c1t0d0s0
2010-06-23.09:30:13 zfs set canmount=noauto rpool
2010-06-23.09:30:13 zfs set mountpoint=/rpool rpool
2010-06-23.09:30:13 zfs create -o mountpoint=legacy rpool/ROOT
2010-06-23.09:30:14 zfs create -b 8192 -V 2048m rpool/swap
2010-06-23.09:30:14 zfs create -b 131072 -V 1024m rpool/dump
2010-06-23.09:30:15 zfs create -o canmount=noauto rpool/ROOT/zfsBE
2010-06-23.09:30:16 zpool set bootfs=rpool/ROOT/zfsBE rpool
2010-06-23.09:30:16 zfs set mountpoint=/ rpool/ROOT/zfsBE
2010-06-23.09:30:16 zfs set canmount=on rpool
2010-06-23.09:30:16 zfs create -o mountpoint=/export rpool/export
2010-06-23.09:30:17 zfs create rpool/export/home
```

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

```
# zpool history -l rpool
History for 'tank':
2010-06-24.13:07:58 zpool create tank mirror c2t2d0 c2t5d0 [user root on neo:global]
```



```

2010-06-24.13:08:23 zpool scrub tank [user root on neo:global]
2010-06-24.13:38:42 zpool clear tank [user root on neo:global]
2010-06-29.11:44:18 zfs create tank/home [user root on neo:global]
2010-06-29.13:28:51 zpool clear tank c2t5d0 [user root on neo:global]
2010-06-30.14:07:40 zpool add tank spare c2t1d0 [user root on neo:global]

```

- L'option `-i` peut être utilisée pour afficher les informations des événements internes à des fins de diagnostic. Exemple :

```
# zpool history -i tank
```

```
History for 'tank':
```

```

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

```

Pour plus d'informations sur l'utilisation de la commande `zpool history`, reportez-vous à la section [“Résolution de problèmes avec le système de fichiers ZFS” à la page 297](#).

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

Solaris 10 10/08 : dans cette version, vous pouvez accorder des autorisations détaillées aux utilisateurs ne disposant d'aucun privilège en vue d'effectuer des tâches d'administration ZFS.

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

Vous pouvez modifier l'administration déléguée avec 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 [Chapitre 9, “Administration déléguée de ZFS”](#) et à la page de manuel [zfs\(1M\)](#).

Configuration de périphériques de journal ZFS distincts

Versión 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. Dans cette version de Solaris, vous pouvez décider si vous souhaitez que les blocs ZIL continuent à être alloués à partir du pool de stockage principal ou à partir d'un autre périphérique de journal. Les performances peuvent parfois être améliorées en vous servant de périphériques de journal d'intention distincts dans votre pool de stockage ZFS, notamment d'une NVRAM ou d'un disque dédié.

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

Vous pouvez installer un périphérique de journalisation ZFS au moment de la création du pool de stockage ou après sa création. Pour obtenir des exemples de configuration de périphériques de journal, reportez-vous aux sections “Création d'un pool de stockage ZFS avec des périphériques de journal” à la page 75 et “Ajout de périphériques à un pool de stockage” à la page 83.

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 journal peuvent être mis en miroir et leur réplication peut être annulée, mais RAID-Z n'est pas pris en charge pour les périphériques de journal.
- Si un périphérique de journal 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 journal peuvent être ajoutés, remplacés, connectés, déconnectés, importés et exportés en tant que partie du pool de stockage. À partir de la version Solaris 10 9/10, les périphériques de journal peuvent être supprimés.
- La taille minimale d'un périphérique de journal 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 journal 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.

Dans l'exemple suivant, les jeux de données ZFS (`users/area51`) sont créés 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 déjà pendant l'opération de création, celle-ci est réussie.

Les propriétés spécifiées s'appliquent au jeu de données cible, mais pas au jeu de données intermédiaire. 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\(1M\)](#).

Amélioration de l'enfichage à chaud ZFS

Solaris 10 10/08 : dans cette version, le système de fichiers ZFS répond plus efficacement aux périphériques qui sont supprimés et peut désormais identifier automatiquement les périphériques insérés.

- 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 le remplacement du périphérique est désactivé, l'administrateur doit l'initier à l'aide de la commande `zpool replace`. Si l'option est définie sur `on`, tout nouveau périphérique se trouvant au même emplacement physique qu'un périphérique qui appartenait auparavant au pool est automatiquement formaté et remplacé. Par défaut, le remplacement est désactivé.
- L'état de pool de stockage `REMOVED` est fourni en cas de retrait physique du périphérique ou d'un disque hot spare alors que le système est en cours d'exécution. Si un disque hot spare est disponible, il remplace le périphérique retiré.
- Si un périphérique est retiré, puis réinséré, il est mis en ligne. Si un disque hot spare est activé lors de la réinsertion du périphérique, le disque hot spare est retiré une fois l'opération en ligne terminée.
- La détection automatique du retrait ou de l'insertion de périphériques dépend du matériel utilisé. Il est possible qu'elle ne soit pas prise en charge sur certaines plates-formes. Par exemple, les périphériques USB sont configurés automatiquement après insertion. Il peut être toutefois nécessaire d'utiliser la commande `cfgadm -c configure` pour configurer un lecteur SATA.

- Les disques hot spare sont consultés régulièrement afin de vérifier qu'ils sont en ligne et disponibles.

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 de manière récursive tous les instantanés ZFS descendants à l'aide de la commande `zfs rename -r`. Exemple :

Dans un premier temps, un instantané d'un jeu de systèmes de fichiers ZFS est créé.

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

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

Ensuite, les instantanés sont renommés le jour suivant.

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

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

Un instantané est le seul type de jeu de données pouvant être renommé de façon récursive.

Pour plus d'informations sur les instantanés, reportez-vous à la section “[Présentation des instantanés ZFS](#)” à la page 225 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

Version 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` 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 206.

Stockage de plusieurs copies de données utilisateur ZFS

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

Cette version vous permet également de 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 support (plus connues sous le nom de *bit rot*) pour l'ensemble des configurations ZFS.
- Cela garantit la sécurité des données, même lorsqu'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.

Remarque – 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 du stockage de plusieurs 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 206.

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 était nécessaire d'utiliser la commande `find -inum` pour identifier les noms de fichiers à partir de la liste des inodes affichés.

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 316.

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, vous pouvez configurer l'initiateur iSCSI. Pour obtenir des informations sur la configuration d'un initiateur Solaris iSCSI, consultez le [Chapitre 14, “Configuring Oracle 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 283.

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 Oracle, d'identifier les commandes ZFS *réelles* 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
```

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*. Cependant, ces informations ne sont enregistrées qu'à partir de la version 10 10/08 de Solaris. Pour plus d'informations, reportez-vous à la section [“Améliorations apportées à l'historique des commandes ZFS \(zpool history\)”](#) à la page 32.

Pour obtenir des informations supplémentaires sur la résolution de problèmes relatifs à ZFS, consultez la section [“Résolution de problèmes avec le système de fichiers ZFS”](#) à la page 297.

Améliorations de propriétés ZFS

Propriété ZFS `xattr`

Version Solaris 10 8/07 : vous pouvez utiliser la propriété `xattr` pour désactiver ou activer les attributs avancés d'un système de fichiers ZFS spécifique. La valeur par défaut est `on`. Pour obtenir une description des propriétés ZFS, reportez-vous à la section [“Présentation des propriétés ZFS”](#) à la page 189.

Propriété ZFS `canmount`

Version 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 201.

Propriétés ZFS définies par l'utilisateur

Version 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 propose 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 202.

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

Version Solaris 10 8/07 : dans cette version de Solaris, vous pouvez définir les propriétés d'un système de fichiers lors de sa création, pas seulement après sa création.

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

Versión 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 spécifiez `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`

Versión 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. Il peut être utile de faire appel à cette option en cas de modification du système de fichiers entre le moment où la restauration se produit et le moment où la réception commence.

Pour plus d'informations, reportez-vous à la section [“Réception d'un instantané ZFS”](#) à la page 236.

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. En outre, vous pouvez utiliser l'option `-r` pour 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 d'opération, une certaine cohérence des données d'instantané est assurée, 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 226.

Configuration RAID-Z à double parité (`raidz2`)

Version 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 d'un pool de stockage RAID-Z” à la page 74 ou la page de manuel `zpool(1M)`.

Disques hot spare pour périphériques de pool de stockage ZFS

Version Solaris 10 11/06 la fonction de disque hot spare ZFS 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 98 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 233 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 ZFS est plus ancienne.

Pour plus d'informations, reportez-vous à la section “Mise à niveau de pools de stockage ZFS” à la page 123 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 déterminer 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 46](#).

Les commandes de sauvegarde et de restauration ZFS sont 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 envoient et reçoivent des 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 234](#).

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 121](#).

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.

Si une panne survient sur un système de fichiers ZFS, un message semblable au message suivant s'affiche :

```
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Wed Jun 30 14:53:39 MDT 2010
PLATFORM: SUNW,Sun-Fire-880, CSN: -, HOSTNAME: neo
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: 504a1188-b270-4ab0-af4e-8a77680576b8
DESC: A ZFS device failed. Refer to http://sun.com/msg/ZFS-8000-D3 for more information.
```

AUTO-RESPONSE: No automated response will occur.
 IMPACT: Fault tolerance of the pool may be compromised.
 REC-ACTION: Run 'zpool status -x' and replace the bad device.

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 suite à un problème de ZFS signalé, consultez la section [“Réparation d'un périphérique manquant”](#) à la page 303.

Commande `zpool clear`

Version Solaris 10 6/06 : cette version inclut la commande `zpool clear`, qui permet d'effacer les décomptes d'erreurs associés à un périphérique ou un 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 section [“Effacement des erreurs de périphérique de pool de stockage”](#) à la page 95 et à la page de manuel `zpool(1M)`.

Format NFSv4 ACL compact

Solaris 10 6/06: dans cette version, vous pouvez définir et afficher les ACL NFSv4 sous deux formats : détaillé et compact. La commande `chmod` permet de définir l'un des formats ACL. Vous pouvez utiliser la commande `ls -V` pour afficher les ACL au format compact. Vous pouvez utiliser la commande `ls -v` pour afficher les ACL au format 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 262, à la page de manuel `chmod(1)` et à la page de manuel `ls(1)`.

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

Version Solaris 10 6/06 : un nouvel outil de contrôle de système de fichiers, `fsstat`, crée des 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 rddir read read write write
file remov chng get set ops ops ops bytes ops bytes
7.82M 5.92M 2.76M 1.02G 3.32M 5.60G 87.0M 363M 1.86T 20.9M 251G zfs
```

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

Gestion Web ZFS

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

- créer un nouveau pool de stockage ;
- augmenter la capacité d'un pool existant ;
- déplacer (exporter) un pool de stockage vers un autre système ;
- importer un pool de stockage précédemment exporté afin qu'il soit disponible dans un autre système ;
- visualiser des informations sur les pools de stockage ;
- créer un système de fichiers ;
- créer un volume ;
- créer un instantané d'un système de fichiers ou d'un volume ;
- restaurer un système de fichiers à l'aide d'un instantané précédent.

Vous pouvez accéder à la console d'administration ZFS par le biais d'un navigateur Web sécurisé en vous connectant à l'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. Le système de fichiers ZFS est un système robuste, évolutif et facile à administrer.

Stockage ZFS mis en pool

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

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

Sémantique transactionnelle

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

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

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

Sommes de contrôle et données d'autorétablissement

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

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

Évolutivité inégalée

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

Instantanés ZFS

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

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

Administration simplifiée

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

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

Terminologie ZFS

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

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

Clone	<p>de somme regroupe entre autres, le contrôle de somme simple et rapide <code>fletcher4</code> (paramètre par défaut), ainsi que les puissantes fonctions de hachage cryptographique telles que <code>SHA256</code>.</p> <p>Système de fichiers dont le contenu initial est identique à celui d'un instantané.</p> <p>Pour plus d'informations sur les clones, reportez-vous à la section “Présentation des clones ZFS” à la page 232.</p>						
Jeu de données	<p>Nom générique pour les composants ZFS suivants : clones, systèmes de fichiers, instantanés et volumes.</p> <p>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 :</p> <p><i>pool/chemin</i> [<i>@instantané</i>]</p> <table><tr><td><i>pool</i></td><td>Identifie le nom d'un pool de stockage contenant le jeu de données.</td></tr><tr><td><i>chemin</i></td><td>Nom de chemin délimité par slash pour le composant de jeu de données</td></tr><tr><td><i>instantané</i></td><td>Composant optionnel identifiant l'instantané d'un jeu de données.</td></tr></table> <p>Pour plus d'informations sur les jeux de données, reportez-vous au Chapitre 6, “Gestion des systèmes de fichiers Oracle Solaris ZFS”.</p>	<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 le composant de jeu de données	<i>instantané</i>	Composant optionnel identifiant l'instantané d'un jeu de données.
<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 le composant de jeu de données						
<i>instantané</i>	Composant optionnel identifiant l'instantané d'un jeu de données.						
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 Chapitre 6, “Gestion des systèmes de fichiers Oracle Solaris ZFS”.</p>						
Miroir	<p>Périphérique virtuel stockant des copies identiques de données sur un ou plusieurs</p>						

	disques. Lorsqu'un disque d'un miroir est défaillant, tout autre disque du miroir est en mesure de fournir les mêmes données.
Pool	<p>Groupe logique de périphériques décrivant la disposition et les caractéristiques physiques du stockage disponible. L'espace disque pour les jeux de données est alloué à partir d'un pool.</p> <p>Pour plus d'informations sur les pools de stockage, reportez-vous au Chapitre 4, "Gestion des pools de stockage Oracle Solaris ZFS".</p>
Environnement d'initialisation principal	Environnement d'initialisation utilisé par la commande <code>lucreate</code> pour créer un environnement d'initialisation alternatif. Par défaut, l'environnement d'initialisation principal correspond à l'environnement d'initialisation actuel. Ce paramètre par défaut peut être modifié à l'aide de l'option <code>lucreate - s</code> .
RAID-Z	Périphérique virtuel stockant les données et la parité sur plusieurs disques. Pour plus d'informations sur RAID-Z, reportez-vous à la section " Configuration de pool de stockage RAID-Z " à la page 69.
Réargenture	<p>Processus de copie de données d'un périphérique à un autre, connu sous le nom de <i>resynchronisation</i>. Par exemple, si un périphérique de miroir est remplacé ou mis hors ligne, les données du périphérique de miroir le plus actuel sont copiées dans le périphérique de miroir nouvellement restauré. Dans les produits de gestion de volumes classiques, ce processus est appelé <i>resynchronisation de miroir</i>.</p> <p>Pour plus d'informations sur la réargenture ZFS, reportez-vous à la section "Affichage de l'état de réargenture" à la page 313.</p>
Instantané	<p>Copie ponctuelle en lecture seule d'un système de fichiers ou d'un volume.</p> <p>Pour plus d'informations sur les instantanés, reportez-vous à la section "Présentation des instantanés ZFS" à la page 225.</p>

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 77.

Volume

Jeu de données représentant un périphérique en mode bloc. Vous pouvez par exemple créer un volume ZFS en tant que périphérique de swap.

Pour plus d'informations sur les volumes ZFS, reportez-vous à la section [“Volumes ZFS”](#) à la page 281.

Exigences d'attribution de noms de composants ZFS

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

- Chaque composant ne peut contenir que des caractères alphanumériques en plus des quatre caractères spéciaux suivants :
 - Soulignement (`_`)
 - Trait d'union (`-`)
 - Deux points (`:`)
 - Point (`.`)
- Les noms de 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.
 - Vous ne pouvez pas utiliser un nom commençant par `mirror`, `raidz`, `raidz1`, `raidz2`, `raidz3` ou `spare` car ces noms sont réservés.
 - Les noms de pools ne doivent pas contenir le signe de pourcentage (`%`).
- Les noms de jeux de données doivent commencer par un caractère alphanumérique.
- Les noms de jeux de données ne doivent pas contenir le signe de pourcentage (`%`).

De plus, les composants vides ne sont pas autorisés.

Mise en route d'Oracle Solaris ZFS

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

Il contient les sections suivantes :

- “Exigences et recommandations en matière de matériel et de logiciel ZFS” à la page 53
- “Création d'un système de fichiers ZFS basique” à la page 54
- “Création d'un pool de stockage ZFS” à la page 55
- “Création d'une hiérarchie de systèmes de fichiers ZFS” à la page 56

Exigences et recommandations en matière de matériel et de logiciel ZFS

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

- Utilisez un système SPARC ou x86 exécutant la version 10 6/06 ou une version plus récente de Solaris.
- L'espace disque minimum requis pour un pool de stockage est de 64 Mo. La taille minimale du disque est de 128 Mo.
- Il est nécessaire de disposer d'au moins 768 Mo de mémoire pour installer un système Solaris. Cependant, pour assurer un fonctionnement correct, vous devez utiliser au moins 1 Go ou plus de mémoire.
- Si vous créez une configuration de disques mise en miroir, utilisez plusieurs contrôleurs.

Création d'un système de fichiers ZFS basique

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

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

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

Pour plus d'informations sur les configurations redondantes de pools ZFS, reportez-vous à la section [“Fonctions de réplication d'un pool de stockage ZFS”](#) à la page 69.

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

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

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

Au sein d'un pool, vous 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 disque disponible que nécessaire et est monté automatiquement sur `/tank/fs`.

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

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

Généralement, vous souhaitez créer et organiser une hiérarchie de systèmes de fichiers correspondant à des besoins spécifiques en matière d'organisation. Pour de plus amples informations sur la création d'une hiérarchie de systèmes de fichiers ZFS, reportez-vous à la section [“Création d'une hiérarchie de systèmes de fichiers ZFS”](#) à la page 56.

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 pour le pool de stockage.

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

Pour l'exemple de stockage utilisé dans la section [“Création d'un pool de stockage ZFS” à la page 56](#), partez du principe que les disques entiers `/dev/dsk/c2t0d0` et `/dev/dsk/c0t1d0` 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 65](#).

2 Sélectionnez la réplication de données.

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

Pour l'exemple de stockage utilisé dans la section [“Création d'un pool de stockage ZFS” à la page 56](#) 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 69](#).

▼ 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 291](#).

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

Le nom sert à identifier le pool de stockage lorsque vous exécutez les commandes `zpool` et `zfs`. 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 52](#).

- 3 **Créez le pool.**

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

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

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

Pour 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 72](#). Pour plus d'informations sur la détection de l'utilisation de périphériques, reportez-vous à la section [“Détection des périphériques utilisés” à la page 78](#).

- 4 **Affichez les résultats.**

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

```
# zpool list
NAME                SIZE  ALLOC  FREE  CAP  HEALTH  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 107](#).

Création d'une hiérarchie de systèmes de fichiers ZFS

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

ZFS permet d'organiser en hiérarchies les systèmes de fichiers. Chaque système de cette hiérarchie ne compte qu'un seul parent. 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. Pour ce faire, nous vous recommandons d'établir un système de fichiers par utilisateur ou par projet car cela permet de contrôler les propriétés, les instantanés et les sauvegardes par utilisateur ou par projet.

Deux systèmes de fichiers ZFS, `bonwick` et `billm` sont créés dans la section [“Création de systèmes de fichiers ZFS” à la page 58](#).

Pour plus d'informations sur la gestion des systèmes de fichiers, reportez-vous au [Chapitre 6, “Gestion des systèmes de fichiers Oracle Solaris ZFS”](#).

2 Regroupez les systèmes de fichiers similaires.

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

Dans l'exemple de la section [“Création de systèmes de fichiers ZFS” à la page 58](#), les deux systèmes de fichiers sont placés sous un système de fichiers appelé `home`.

3 Choisissez les propriétés du système de fichiers.

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

Dans l'exemple de la section [“Création de systèmes de fichiers ZFS” à la page 58](#), tous les répertoires de base sont montés dans `/export/zfs/user`. Ils sont partagés à l'aide de NFS et la compression est activée. De plus, un quota de 10 Go est appliqué à l'utilisateur `bonwick`.

Pour de plus amples informations sur les propriétés, reportez-vous à la section [“Présentation des propriétés ZFS” à la page 189](#).

▼ 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 291](#).

- 2 **Créez la hiérarchie souhaitée.**

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

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

- 3 **Définissez les propriétés héritées.**

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

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

Il est possible de définir les propriétés du système de fichiers lors de la création de ce dernier. 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 189](#).

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

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

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

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

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

Pour de plus amples informations sur les systèmes de fichiers, reportez-vous à la section [“Création d'un système de fichiers ZFS” à la page 186](#).

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 211.

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 204.

Pour de plus amples informations sur l'utilisation et le calcul de l'espace disque, reportez-vous à la section [“Comptabilisation de l'espace disque ZFS”](#) à la page 62.

Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques

Ce chapitre aborde les différences significatives entre Oracle Solaris ZFS et les systèmes de fichiers classiques. Lors de l'utilisation d'outils classiques avec le système de fichiers ZFS, la compréhension de ces différences clés permet d'éviter les confusions.

Il contient les sections suivantes :

- “Granularité du système de fichiers ZFS” à la page 61
- “Comptabilisation de l'espace disque ZFS” à la page 62
- “Comportement d'espace saturé” à la page 62
- “Montage de système de fichiers ZFS” à la page 63
- “Gestion de volumes classique” à la page 63
- “Nouveau modèle ACL Solaris” à la page 63

Granularité du système de fichiers ZFS

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

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

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

Pour obtenir un exemple de création d'une hiérarchie de système de fichiers, reportez-vous à la section “Création d'une hiérarchie de systèmes de fichiers ZFS” à la page 56.

Comptabilisation de l'espace disque ZFS

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

Notez que la taille maximale du système de fichiers peut être limitée par l'utilisation des quotas. Pour obtenir des informations sur les quotas, reportez-vous à la section “[Définitions de quotas sur les systèmes de fichiers ZFS](#)” à la page 219. Vous pouvez allouer une certaine quantité d'espace disque à un système de fichiers à l'aide des réservations. Pour obtenir des informations sur les réservations, reportez-vous à la rubrique “[Définition de réservations sur les systèmes de fichiers ZFS](#)” à la page 222. Ce modèle est très similaire au modèle NFS dans lequel plusieurs répertoires sont montés à partir du même système de fichiers (par exemple : `/home`).

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

ZFS est un système de fichiers transactionnel. La plupart des modifications apportées au système de 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 disque 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 disque ne sont pas automatiquement mises à jour.

Pour plus de détails sur la consommation d'espace ZFS telle que signalée par les commandes du et `df`, reportez-vous à la page suivante :

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

Comportement d'espace saturé

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

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

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

Montage de système de fichiers ZFS

Le système de fichiers ZFS réduit la complexité et facilite l'administration. Par exemple, avec des systèmes de fichiers standard, vous devez modifier le fichier `/etc/vfstab` à chaque fois que vous ajoutez un système de fichiers. Avec ZFS, cela n'est plus nécessaire, grâce au montage et démontage automatique en fonction des propriétés du 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 211.

Gestion de volumes classique

Comme décrit à la section [“Stockage ZFS mis en pool”](#) à la page 47, 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 :

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

Pour plus d'informations sur l'utilisation de listes de contrôle d'accès (ACL) avec des fichiers ZFS, reportez-vous au [Chapitre 8, "Utilisation des ACL pour protéger les fichiers Oracle Solaris ZFS"](#).

Gestion des pools de stockage Oracle Solaris ZFS

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

Il contient les sections suivantes :

- “Composants d'un pool de stockage ZFS” à la page 65
- “Fonctions de réplication d'un pool de stockage ZFS” à la page 69
- “Création et destruction de pools de stockage ZFS” à la page 72
- “Gestion de périphériques dans un pool de stockage ZFS” à la page 82
- “Gestion des propriétés de pool de stockage ZFS” à la page 104
- “Requête d'état de pool de stockage ZFS” à la page 107
- “Migration de pools de stockage ZFS” à la page 116
- “Mise à niveau de pools de stockage ZFS” à la page 123

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 65
- “Utilisation de tranches dans un pool de stockage ZFS” à la page 67
- “Utilisation de fichiers dans un pool de stockage ZFS” à la page 68

Utilisation de disques dans un pool de stockage ZFS

Le composant le plus basique d'un pool de stockage est le stockage physique. Le stockage physique peut être constitué de tout périphérique en mode bloc d'une taille supérieure à 128 Mo. En 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. La manière d'opérer recommandée consiste à utiliser un disque entier. Dans ce cas, il est inutile de formater spécifiquement le disque. ZFS formate le disque à l'aide d'une étiquette EFI de façon à ce qu'il contienne une grande tranche unique. Utilisé de cette façon, le tableau de partition affiché par la commande `format` s'affiche comme suit :

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

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

Pour utiliser un disque entier, celui-ci doit être nommé en utilisant la convention de dénomination `/dev/dsk/cXtXdX`. 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. Pour plus d'informations sur les étiquettes EFI, reportez-vous à la section [“EFI Disk Label” du *System Administration Guide: Devices and File Systems*](#).

Un disque destiné à un pool racine ZFS doit être créé avec une étiquette SMI et non une étiquette EFI. Vous pouvez réattribuer une étiquette SMI à un disque à l'aide de la commande `format - e`.

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

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

L'utilisation de disques physiques constitue la méthode de création de pools de stockage ZFS la plus simple. Les configurations ZFS deviennent de plus en plus complexes, en termes de gestion, de fiabilité et de performance. Lorsque vous construisez des pools à partir de tranches de disques, de LUN dans des baies RAID matérielles ou de volumes présentés par des gestionnaires de volume basés sur des logiciels. Les considérations suivantes peuvent vous aider à configurer ZFS avec d'autres solutions de stockage matérielles ou logicielles :

- Si vous élaborez une configuration ZFS sur des LUN à partir de baies RAID matérielles, vous devez comprendre la relation entre les fonctionnalités de redondance ZFS et les fonctionnalités de redondance proposées par la baie. Certaines configurations peuvent fournir une redondance et des performances adéquates, mais d'autres non.
- Vous pouvez construire des périphériques logiques pour ZFS à l'aide de volumes présentés par des gestionnaires de volumes logiciels tels que Solaris Volume Manager (SVM) ou Veritas Volume Manager (VxVM). Ces configurations sont cependant déconseillées. Même si le système de fichiers ZFS fonctionne correctement sur ces périphériques, il se peut que les performances ne soient pas optimales.

Pour 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. Pour les systèmes sur lesquels les informations de l'ID du périphérique sont disponibles, cette méthode d'identification permet de reconfigurer les périphériques sans mettre à jour ZFS. Étant donné que la génération et la gestion d'ID de périphérique peuvent varier d'un système à l'autre, vous devez commencer par exporter le pool avant tout déplacement de périphériques, par exemple, le déplacement d'un disque d'un contrôleur à un autre. Un événement système, tel que la mise à jour du microprogramme ou toute autre modification apportée au matériel, peut modifier les ID de périphérique du pool de stockage ZFS, ce qui peut entraîner l'indisponibilité des périphériques.

Utilisation de tranches dans un pool de stockage ZFS

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

Pour un pool racine ZFS initialisable, les disques du pool doivent contenir des tranches et doivent être étiquetés avec une étiquette SMI. 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.

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

```
# format
.
.
.
Specify disk (enter its number): 4
selecting cltld0
partition> p
Current partition table (original):
Total disk cylinders available: 14087 + 2 (reserved cylinders)
```

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

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

format

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

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

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 configuration complexe, lorsque les périphériques physiques présents ne sont pas suffisants. Tous les fichiers doivent être spécifiés avec leur chemin complet et leur taille doit être de 64 Mo minimum.

Fonctions de réplication d'un pool de stockage ZFS

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

- “Configuration de pool de stockage mis en miroir” à la page 69
- “Configuration de pool de stockage RAID-Z” à la page 69
- “Données d'autorétablissement dans une configuration redondante” à la page 71
- “Entrelacement dynamique dans un pool de stockage” à la page 71
- “Pool de stockage ZFS hybride” à la page 71

Configuration de pool de stockage mis en miroir

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

```
mirror c1t0d0 c2t0d0
```

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

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

Pour obtenir des informations sur les pools de stockage mis en miroir, reportez-vous à la section “Création d'un pool de stockage mis en miroir” à la page 72.

Configuration de pool de stockage RAID-Z

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

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

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

Tous les algorithmes similaires à RAID-5 (RAID-4, RAID-6, RDP et EVEN-ODD, par exemple) peuvent souffrir d'un problème connu sous le nom de "RAID-5 write hole". Si seule une partie d'un entrelacement RAID-5 est écrite, et qu'une perte d'alimentation se produit avant que tous les blocs aient été écrits sur le disque, la parité n'est pas synchronisée avec les données, et est par conséquent inutile à tout jamais (à moins qu'elle ne soit écrasée ultérieurement par une écriture d'entrelacement total). Dans RAID-Z, ZFS utilise des entrelacements RAID de largeur variable

pour que toutes les écritures correspondent à des entrelacements entiers. Cette conception n'est possible que parce que ZFS intègre le système de fichiers et la gestion de périphérique de telle façon que les métadonnées du système de fichiers disposent de suffisamment d'informations sur le modèle de redondance de données pour gérer les entrelacements RAID de largeur variable. RAID-Z est la première solution au monde pour le trou d'écriture de RAID-5.

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

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

```
raidz c1t0d0 c2t0d0 c3t0d0
```

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

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

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

Pour obtenir des informations sur les pools de stockage RAID-Z, reportez-vous à la section [“Création d'un pool de stockage RAID-Z”](#) à la page 74.

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

http://blogs.sun.com/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

Pool de stockage ZFS hybride

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

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

Données d'autorétablissement dans une configuration redondante

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

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

Entrelacement dynamique dans un pool de stockage

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

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

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

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

Création et destruction de pools de stockage ZFS

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

- [“Création d'un pool de stockage ZFS” à la page 72](#)
- [“Affichage des informations d'un périphérique virtuel de pool de stockage” à la page 77](#)
- [“Gestion d'erreurs de création de pools de stockage ZFS” à la page 78](#)
- [“Destruction de pools de stockage ZFS” à la page 81](#)

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

Création 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 52](#).

Création d'un pool de stockage de base

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

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

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

Création d'un pool de stockage mis en miroir

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

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

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

Pour plus d'informations sur les configurations 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 périphérique virtuel (vdev) supplémentaire de niveau supérieur à une configuration en miroir existante. Pour plus d'informations, reportez-vous à la rubrique “Ajout de périphériques à un pool de stockage” à la page 83.
- 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 88.
- Remplacement d'un ou de plusieurs disques dans une configuration en miroir existante, à condition que les disques de remplacement soient d'une taille supérieure ou égale à celle du périphérique remplacé. Pour plus d'informations, reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 96.
- Retrait d'un ou de plusieurs disques dans une configuration en miroir, à condition que les périphériques restants procurent la redondance qui convient à la configuration. Pour plus d'informations, reportez-vous à la section “Connexion et séparation de périphériques dans un pool de stockage” à la page 88.
- Scission d'une configuration mise en miroir en déconnectant l'un des disques en vue de créer un nouveau pool identique. Pour plus d'informations, reportez-vous à la section “Création d'un pool par scission d'un pool de stockage ZFS mis en miroir” à la page 89.

Vous ne pouvez pas forcer la suppression d'un périphérique qui n'est pas un périphérique de journal ni de cache d'un pool de stockage mis en miroir. Cette fonction fait l'objet d'une demande d'amélioration.

Création d'un pool racine ZFS

Vous pouvez installer et initialiser un système de fichiers racine ZFS. Vérifiez les informations de configuration du pool racine :

- Les disques utilisés pour le pool racine doivent avoir une étiquette VTOC (SMI) et le pool doit être créé avec des tranches de disque.
- Le pool racine doit être créé sous la forme d'une configuration en miroir ou d'une configuration à disque unique. Vous ne pouvez pas ajouter d'autres disques mis en miroir pour créer plusieurs périphériques virtuels de niveau supérieur à l'aide de la commande `zpool add`. Toutefois, vous pouvez étendre un périphérique virtuel mis en miroir à l'aide de la commande `zpool attach`.
- Les configurations RAID-Z ou entrelacées ne sont pas prises en charge.

- Un pool racine ne peut pas avoir de périphérique de journal distinct.
- Si vous tentez d'utiliser une configuration non prise en charge pour un pool racine, un message tel que le suivant s'affiche :

```
ERROR: ZFS pool <pool-name> does not support boot environments
# zpool add -f rpool log c0t6d0s0
cannot add to 'rpool': root pool can not have multiple vdevs or separate logs
```

Pour plus d'informations sur l'installation et l'initialisation d'un système de fichiers racine ZFS, reportez-vous [Chapitre 5, "Installation et initialisation d'un système de fichiers racine Oracle Solaris ZFS"](#).

Création d'un pool de stockage RAID-Z

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

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

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

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

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

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

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

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

config:

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

errors: No known data errors

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

- Ajout d'un autre jeu de disques comme périphérique virtuel (vdev) supplémentaire de niveau supérieur à une configuration RAID-Z existante. Pour plus d'informations, reportez-vous à la rubrique [“Ajout de périphériques à un pool de stockage”](#) à la page 83.
- 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 96.

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

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

Pour obtenir des informations supplémentaire, reportez-vous à la section [“Configuration de pool de stockage RAID-Z”](#) à la page 69.

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

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 journal ZFS, reportez-vous à la section [“Configuration de périphériques de journal ZFS distincts”](#) à la page 34.

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

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

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

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

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

Pour plus d'informations sur la récupération suite à une défaillance de périphérique de journal, reportez-vous à l'[Exemple 11-2](#).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Gestion d'erreurs de création de pools de stockage ZFS

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

Détection des périphériques utilisés

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

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

`/dev/dsk/c1t0d0s1` is currently mounted on swap. Please see `swap(1M)`.
`/dev/dsk/c1t1d0s0` is part of active ZFS pool `zpool`. Please see `zpool(1M)`.

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

Système de fichiers monté	Le disque ou une de ses tranches contient un système de fichiers actuellement monté. La commande <code>umount</code> permet de corriger cette erreur.
Système de fichiers dans <code>/etc/vfstab</code>	Le disque contient un système de fichiers répertorié dans le fichier <code>/etc/vfstab</code> , mais le système de fichiers n'est pas monté. Pour corriger cette erreur, supprimez ou commentez la ligne dans le fichier <code>/etc/vfstab</code> .
Périphérique de vidage dédié	Le disque est utilisé en tant que périphérique de vidage dédié pour le système. La commande <code>dumpadm</code> 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 <code>zpool destroy</code> afin de détruire l'autre pool s'il est obsolète. Utilisez sinon la commande <code>zpool detach</code> 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 Solaris Volume Manager.
Live upgrade	Le disque est en cours d'utilisation en tant qu'environnement d'initialisation de remplacement pour Oracle Solaris Live Upgrade.
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` pour les ignorer.

Niveaux de réplication incohérents

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

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

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

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

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

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

    tank
    mirror
```



```
c1t0d0
c1t1d0
```

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

Point de montage par défaut pour les pools de stockage

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

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

Cette commande crée le pool `home` et le 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 211](#).

Destruction de pools de stockage ZFS

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

```
# zpool destroy tank
```



Attention – Faites très attention lorsque vous détruisez un pool. Assurez-vous de détruire le pool souhaité et de toujours disposer de copies de vos données. En cas de destruction accidentelle d'un pool, vous pouvez tenter de le récupérer. Pour obtenir des informations supplémentaires, reportez-vous à la section [“Récupération de pools de stockage ZFS détruits” à la page 121](#).

Destruction d'un pool avec des périphériques défectueux

La destruction d'un pool requiert l'écriture des données sur le disque pour indiquer que le pool n'est désormais plus valide. Ces informations d'état évitent que les périphériques ne s'affichent

en tant que pool potentiel lorsque vous effectuez une importation. La destruction du pool est tout de même possible si un ou plusieurs périphériques ne sont pas disponibles. Cependant, les informations d'état requises ne sont pas écrites sur ces périphériques indisponibles.

Ces périphériques, lorsqu'ils sont correctement réparés, sont signalés comme *potentiellement actifs*, lors de la création d'un pool. Lorsque vous recherchez des pools à importer, ils s'affichent en tant que périphériques valides. Si un pool a tant de périphérique défaillants 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. 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 113.

Pour de plus amples informations sur l'importation de pools, reportez-vous à la section [“Importation de pools de stockage ZFS”](#) à la page 120.

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 65. Après la création d'un pool, vous pouvez effectuer plusieurs tâches de gestion des périphériques physiques au sein du pool.

- [“Ajout de périphériques à un pool de stockage”](#) à la page 83
- [“Connexion et séparation de périphériques dans un pool de stockage ”](#) à la page 88
- [“Création d'un pool par scission d'un pool de stockage ZFS mis en miroir”](#) à la page 89
- [“Mise en ligne et mise hors ligne de périphériques dans un pool de stockage”](#) à la page 93
- [“Effacement des erreurs de périphérique de pool de stockage”](#) à la page 95
- [“Remplacement de périphériques dans un pool de stockage”](#) à la page 96
- [“Désignation des disques hot spare dans le pool de stockage”](#) à la page 98

Ajout de périphériques à un pool de stockage

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

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

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

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

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

Pour plus d'informations sur la validation des périphériques virtuels, reportez-vous à la section [“Détection des périphériques utilisés”](#) à la page 78.

EXEMPLE 4-1 Ajout de disques à une configuration ZFS mise en miroir

Dans l'exemple suivant, un autre miroir est ajouté à une configuration ZFS en miroir existante sur un système Oracle 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-0	ONLINE	0	0	0
c0t1d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
mirror-1	ONLINE	0	0	0
c0t2d0	ONLINE	0	0	0

EXEMPLE 4-1 Ajout de disques à une configuration ZFS mise en miroir (Suite)

```

c1t2d0 ONLINE      0    0    0

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

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

errors: No known data errors

```

EXEMPLE 4-2 Ajout de disques à une configuration RAID-Z

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

```

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

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

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

NAME      STATE      READ WRITE CKSUM

```

EXEMPLE 4-2 Ajout de disques à une configuration RAID-Z (Suite)

```

rzpool      ONLINE      0      0      0
raidz1-0    ONLINE      0      0      0
c1t0d0     ONLINE      0      0      0
c1t2d0     ONLINE      0      0      0
c1t3d0     ONLINE      0      0      0
raidz1-1    ONLINE      0      0      0
c2t2d0     ONLINE      0      0      0
c2t3d0     ONLINE      0      0      0
c2t4d0     ONLINE      0      0      0

```

errors: No known data errors

EXEMPLE 4-3 Ajout et suppression d'un périphérique de journal mis en miroir

L'exemple suivant illustre l'ajout d'un périphérique de journal en miroir à un pool de stockage en miroir. Pour plus d'informations sur l'utilisation de périphériques de journal dans votre pool de stockage, reportez-vous à la section [“Configuration de périphériques de journal ZFS distincts”](#) à la page 34.

```

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

```

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

errors: No known data errors

```

# zpool add newpool log mirror c0t6d0 c0t7d0
# zpool status newpool
pool: newpool
state: ONLINE
scrub: none requested
config:

```

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

errors: No known data errors

EXEMPLE 4-3 Ajout et suppression d'un périphérique de journal mis en miroir (Suite)

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.

Les périphériques de journal peuvent être supprimés à l'aide de la commande `zpool remove`. Le périphérique de journal mis en miroir dans l'exemple précédent peut être supprimé en spécifiant l'argument `mirror-1`. Exemple :

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

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

errors: No known data errors
```

Si la configuration de votre pool contient uniquement un périphérique de journal, vous devez le supprimer en spécifiant le nom de ce dernier. Exemple :

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

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

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

EXEMPLE 4-4 Ajout et suppression des périphériques de cache

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

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

EXEMPLE 4-4 Ajout et suppression des périphériques de cache (Suite)

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

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

errors: No known data errors
```

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

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

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

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

errors: No known data errors
```

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

Pour plus d'informations sur l'utilisation des périphériques de cache dans un pool de stockage ZFS, reportez-vous à la section [“Création d'un pool de stockage ZFS avec des périphériques de cache”](#) à la page 76.

Connexion et séparation de périphériques dans un pool de stockage

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

Si vous connectez un disque pour créer un pool racine mis en miroir, reportez-vous à la section “Création d’un pool racine mis en miroir (post-installation)” à la page 136.

Si vous remplacez un disque dans le pool racine ZFS, reportez-vous à la section “Remplacement d’un disque dans le pool racine ZFS” à la page 178.

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

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

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

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

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

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

errors: No known data errors
```

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

EXEMPLE 4-6 Conversion d'un pool de stockage ZFS non redondant en pool de stockage ZFS en miroir

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

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

errors: No known data errors
# zpool attach tank c0t1d0 c1t1d0
# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Fri Jan  8 14:28:23 2010
config:
    NAME          STATE          READ WRITE CKSUM
    tank          ONLINE         0     0     0
    mirror-0     ONLINE         0     0     0
    c0t1d0       ONLINE         0     0     0
    c1t1d0       ONLINE         0     0     0  73.5K resilvered

errors: No known data errors
```

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

```
# zpool detach zeepool c2t1d0
```

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

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

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

Vous pouvez rapidement cloner un pool de stockage ZFS mis en miroir en tant que pool de sauvegarde avec la commande `zpool split`.

Cette fonctionnalité ne peut actuellement pas être utilisée pour scinder un pool racine mis en miroir.

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

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

Exemple :

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

    NAME          STATE          READ WRITE CKSUM
    tank          ONLINE         0     0     0
      mirror-0    ONLINE         0     0     0
        c1t0d0    ONLINE         0     0     0
        c1t2d0    ONLINE         0     0     0
```

errors: No known data errors

```
# zpool split tank tank2
```

```
# zpool import tank2
```

```
# zpool status tank tank2
```

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

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

errors: No known data errors

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

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

errors: No known data errors

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

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

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

Si le pool à scinder ne contient aucun point de montage de jeu de données par défaut et que le nouveau pool est créé sur le même système, vous devrez utiliser l'option `zpool split -R` pour identifier un autre répertoire racine pour le nouveau pool afin d'éviter tout conflit entre les points de montage existants le cas échéant. Exemple :

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

Si vous n'utilisez pas l'option `zpool split -R` et que des points de montage entrent en conflit lorsque vous tentez d'importer le nouveau pool, importez celui-ci à l'aide de l'option `-R`. Si le nouveau pool est créé sur un autre système, vous ne devez pas spécifier un autre répertoire racine, sauf en cas de conflits entre les points de montage.

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

- Cette fonction n'est pas disponible dans une configuration RAID-Z ou un pool non redondant composé de plusieurs disques.
- Avant de tenter une opération `zpool split`, les opérations des données et des applications doivent être suspendues.
- Si vous voulez que les disques effectuent les opérations demandées et qu'ils ne les ignorent pas, la commande de mise en cache des enregistrements de vidage des disques est importante.
- Vous ne pouvez pas scinder un pool si une réargenture est en cours.
- Lorsqu'un pool mis en miroir est composé de deux à trois disques dans lesquels le dernier disque du pool d'origine est utilisé pour le nouveau pool créé, la meilleure solution consiste à scinder le pool mis en miroir. Vous pouvez ensuite utiliser la commande `zpool attach` pour recréer votre pool de stockage d'origine mis en miroir ou convertir votre nouveau pool dans un pool de stockage mis en miroir. Il n'existe actuellement aucun moyen de créer un *nouveau* pool mis en miroir à partir d'un des pools mis en miroir *existants* en utilisant cette fonction.
- Si le pool existant est un miroir tri-directionnel, le nouveau pool contiendra un disque après l'opération de scission. Si le pool existant est un miroir bi-directionnel composé de deux disques, cela donne deux pools non redondants composés de deux disques. Vous devez connecter deux disques supplémentaires pour convertir les pools non redondants en pools mis en miroir.
- Pour conserver vos données redondantes lors d'une scission, scindez un pool de stockage mis en miroir composé de trois disques pour que le pool d'origine soit composé de deux disques après la scission.

EXEMPLE 4-7 Scission d'un pool ZFS mis en miroir

Dans l'exemple suivant, un pool de stockage mis en miroir nommé `trinity` et contenant trois disques (`c1t0d0`, `c1t2d0` et `c1t3d0`) est scindé. Les deux pools correspondants sont le pool mis en miroir `trinity` contenant les disques `c1t0d0` et `c1t2d0` et le nouveau pool `neo` contenant le disque `c1t3d0`. Chaque pool contient les mêmes données.

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

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

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

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

errors: No known data errors

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

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

errors: No known data errors
```

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

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

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

Vous pouvez utiliser la commande `zpool 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 resynchronisation 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
```

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

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

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

- Par défaut, l'état `OFFLINE` 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 96.

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 107.

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 113.

Mise en ligne d'un périphérique

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

```
# 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, le remplacez, 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 :

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

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

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

Pour obtenir des informations sur le remplacement d'un périphérique défaillant, reportez-vous à la section [“Réparation d'un périphérique manquant”](#) à la page 303.

Vous pouvez utiliser la commande `zpool online -e` pour étendre un LUN. Par défaut, un LUN ajouté à un pool n'est pas étendu à sa taille maximale, à moins que la propriété `autexpand` du pool ne soit activée. Vous pouvez étendre automatiquement le LUN en utilisant la commande `zpool online -e`, même si le LUN est déjà en ligne ou s'il est actuellement hors ligne. Exemple :

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

Effacement des erreurs de périphérique de pool de stockage

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

Si elle est spécifiée sans argument, cette commande efface toutes les erreurs de périphérique dans le pool. 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 307.

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. Le système de fichiers ZFS reconnaît que le périphérique est un disque différent dans le même emplacement d'un composant matériel. Par exemple, pour remplacer un disque défaillant (`c1t1d0`), supprimez-le, puis ajoutez le disque de rechange au même emplacement, à l'aide de la syntaxe suivante :

```
# zpool replace tank c1t1d0
```

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

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

Si vous remplacez un disque dans le pool racine ZFS, reportez-vous à la section [“Remplacement d'un disque dans le pool racine ZFS”](#) à la page 178.

Voici les étapes de base pour remplacer un disque :

- 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 pouvoir le mettre hors ligne. Si vous remplacez un disque dans le même emplacement sur ce système, vous pouvez exécuter la commande `zpool replace` comme décrit dans le premier exemple de cette section.

Pour obtenir un exemple de remplacement d'un disque sur un système Sun Fire X4500, reportez-vous à l'[Exemple 11-1](#).

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

- Si vous définissez la propriété de pool `autoreplace` sur `on`, tout nouveau périphérique détecté au même emplacement physique qu'un périphérique appartenant précédemment au pool est automatiquement formaté et remplacé. Lorsque cette propriété est activée, vous n'êtes pas obligé d'utiliser la commande `zpool replace`. Cette fonction n'est pas disponible sur tous les types de matériel.
- La taille du périphérique de remplacement doit être égale ou supérieure au disque le plus petit d'une configuration RAID-Z ou mise en miroir.
- Lorsqu'un périphérique de remplacement dont la taille est supérieure à la taille du périphérique qu'il remplace est ajouté à un pool, ce dernier n'est pas automatiquement étendu à sa taille maximale. La valeur de la propriété `autoexpand` du pool détermine si un LUN de remplacement est étendu à sa taille maximale lorsque le disque est ajouté au pool. Par défaut, la propriété `autoexpand` est désactivée. Vous pouvez activer cette propriété pour augmenter la taille du LUN avant ou après avoir ajouté le plus grand LUN au pool.

Dans l'exemple suivant, deux disques de 16 Go d'un pool mis en miroir sont remplacés par deux disques de 72 Go. La propriété `autoexpand` est activée après le remplacement du disque pour étendre le LUN à sa taille maximale.

```
# zpool create pool mirror c1t16d0 c1t17d0
# zpool status
  pool: pool
  state: ONLINE
  scrub: none requested
  config:

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

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

- Le remplacement d'un grand nombre de disques dans un pool volumineux prend du temps, en raison de la resynchronisation des données sur les nouveaux disques. En outre, il peut s'avérer utile d'exécuter la commande `zpool scrub` entre chaque remplacement de disque afin de garantir le fonctionnement des périphériques de remplacement et l'exactitude des données écrites.
- Si un disque défectueux a été remplacé automatiquement par un disque hot spare, il se peut que vous deviez déconnecter le disque hot spare une fois le disque défectueux remplacé. Pour plus d'informations sur la déconnexion d'un disque hot spare, reportez-vous à la section “Activation et désactivation de disque hot spare dans le pool de stockage” à la page 100.

Pour plus d'informations sur le remplacement de périphériques, reportez-vous aux sections “Réparation d'un périphérique manquant” à la page 303 et “Remplacement ou réparation d'un périphérique endommagé” à la page 305.

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. Cependant, ils ne peuvent pas être partagés entre plusieurs pools sur plusieurs systèmes.

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

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

NAME	STATE	READ	WRITE	CKSUM
trinity	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0

```

        c2t1d0 ONLINE      0    0    0
spares
  c1t2d0  AVAIL
  c2t2d0  AVAIL

```

errors: No known data errors

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

```

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

```

```

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

```

errors: No known data errors

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

```

# zpool remove zeepool c2t3d0
# zpool status zeepool
pool: zeepool
state: ONLINE
scrub: none requested
config:

```

```

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

```

errors: No known data errors

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

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

- Actuellement, la commande `zpool remove` ne peut être utilisée que pour la suppression de disques hot spare, de périphériques de journal et de périphériques de cache.

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

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

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

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

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

La commande `zpool replace` permet de remplacer manuellement un périphérique par un disque hot spare. Reportez-vous à l'[Exemple 4-8](#).

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-8000-2Q
scrub: resilver completed after 0h0m with 0 errors on Mon Jan 11 10:20:35 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
zeepool	DEGRADED	0	0	0

```

mirror-0    DEGRADED    0    0    0
c1t2d0     ONLINE         0    0    0
spare-1    DEGRADED    0    0    0
  c2t1d0    UNAVAIL      0    0    0  cannot open
  c2t3d0    ONLINE       0    0    0  88.5K resilvered
spares
c2t3d0     INUSE        currently in use

```

errors: No known data errors

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

- en supprimant le disque hot spare du pool de stockage ;
- en déconnectant le disque hot spare après avoir remplacé physiquement un disque défectueux ; Reportez-vous à l'[Exemple 4–9](#).
- en swappant de manière temporaire ou permanente les données vers le disque hot spare ; Reportez-vous à l'[Exemple 4–10](#).

EXEMPLE 4–8 en remplaçant manuellement un disque par un disque hot spare.

Dans cet exemple, la commande `zpool replace` est utilisée pour remplacer le disque `c2t1d0` par un disque hot spare `c2t3d0`.

```

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

```

```

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

```

errors: No known data errors

Déconnectez ensuite le disque `c2t1d0`.

```

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

```

```

NAME          STATE      READ WRITE CKSUM

```

EXEMPLE 4-8 en remplaçant manuellement un disque par un disque hot spare. (Suite)

```
zeepool    ONLINE      0    0    0
mirror-0   ONLINE      0    0    0
  c1t2d0   ONLINE      0    0    0
  c2t3d0   ONLINE      0    0    0 90K resilvered
```

errors: No known data errors

EXEMPLE 4-9 Déconnexion d'un disque hot spare après le remplacement du disque défectueux

Dans cet exemple, le disque défectueux (c2t1d0) est remplacé physiquement et ZFS est averti à l'aide de la commande `zpool replace`.

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

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

errors: No known data errors

Vous pouvez ensuite utiliser la commande `zpool detach` pour retourner le disque hot spare au pool de disques hot spare. Exemple :

```
# zpool detach zeepool c2t3d0
# zpool status zeepool
pool: zeepool
state: ONLINE
scrub: resilver completed with 0 errors on Wed Jan 20 10:08:44 2010
config:
```

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

errors: No known data errors

EXEMPLE 4-10 Déconnexion d'un disque défectueux et utilisation d'un disque hot spare

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

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

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

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

```
# zpool detach zeepool c2t1d0
```

```
# zpool status zeepool
```

```
pool: zeepool
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 13:46:46 2010
config:
```

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

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

```
(Original failed disk c2t1d0 is physically replaced)
```

```
# zpool add zeepool spare c2t1d0
```

```
# zpool status zeepool
```

```
pool: zeepool
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 13:48:46 2010
config:
```

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

EXEMPLE 4-10 Déconnexion d'un disque défectueux et utilisation d'un disque hot spare (Suite)

```
c2t1d0    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
pool size 68G -
pool capacity 0% -
pool altroot - default
pool health ONLINE -
pool guid 601891032394735745 default
pool version 22 default
pool bootfs - default
pool delegation on default
pool autoreplace off default
pool cachefile - default
pool failmode wait default
pool listsnapshots on default
pool autoexpand off default
pool free 68.0G -
pool allocated 76.5K -
```

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 4-1 Description des propriétés d'un pool ZFS

Nom de la propriété	Type	Valeur par défaut	Description
allocated	Chaîne	SO	Valeur en lecture seule permettant d'identifier l'espace de stockage disponible physiquement alloué dans le pool.

TABLEAU 4-1 Description des propriétés d'un pool ZFS (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
altroot	Chaîne	off	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.
autoreplace	Booléen	off	Contrôle le remplacement automatique d'un périphérique. Si la valeur off est définie, le remplacement du périphérique doit être initié à l'aide de la commande <code>zpool replace</code> . Si la valeur est définie sur on, tout nouveau périphérique se trouvant au même emplacement physique qu'un périphérique qui appartenait au pool est automatiquement formaté et remplacé. L'abréviation de la propriété est la suivante : <code>replace</code> .
bootfs	Booléen	SO	Identifie le jeu de données d'initialisation par défaut du pool racine. Cette propriété est généralement définie par les programmes d'installation et de mise à niveau.
cachefile	Chaîne	SO	Contrôle l'emplacement de la mise en cache du 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 pour mettre en cache les informations de configuration du pool dans un autre emplacement. Ces informations peuvent être importées ultérieurement à l'aide de la commande <code>zpool import -c</code> . Pour la plupart des configurations ZFS, cette propriété n'est pas utilisée.
capacity	Valeur numérique	SO	Valeur en lecture seule identifiant le pourcentage d'espace utilisé du pool. L'abréviation de la propriété est <code>cap</code> .
delegation	Booléen	on	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 Chapitre 9, "Administration déléguée de ZFS" .

TABLEAU 4-1 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 grave d'un pool. Cette condition résulte habituellement d'une perte de connectivité aux périphériques de stockage sous-jacents ou d'une panne de tous les périphériques au sein du pool. Le comportement d'un événement de ce type est déterminé par l'une des valeurs suivantes :</p> <ul style="list-style-type: none"> ■ <code>wait</code> : bloque toutes les requête d'E/S vers le pool jusqu'au rétablissement de la connectivité et jusqu'à l'effacement des erreurs à l'aide de la commande <code>zpool clear</code>. Dans cet état, les opérations d'E/S du pool sont bloquées mais les opérations de lecture peuvent aboutir. Un pool renvoie l'état <code>wait</code> jusqu'à ce que le problème du périphérique soit résolu. ■ <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> : affiche un message sur la console et génère un vidage sur incident du système.
<code>free</code>	Chaîne	SO	Valeur en lecture seule identifiant le nombre de blocs non alloués au sein du pool.
<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 l'état actuel du pool ; les valeurs possibles sont : ONLINE, DEGRADED, FAULTED, OFFLINE, REMOVED ou UNAVAIL.
<code>listsnapshots</code>	Chaîne	on	Détermine si les informations sur les instantanés associées à ce groupe s'affichent avec la commande <code>zfs list</code> . Si cette propriété est désactivée, les informations sur les instantanés peuvent être affichées à l'aide de la commande <code>zfs list -t snapshot</code> .
<code>size</code>	Valeur numérique	SO	Propriété en lecture seule identifiant la taille totale du pool de stockage.

TABLEAU 4-1 Description des propriétés d'un pool ZFS (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
version	Valeur numérique	SO	Identifie la version actuelle sur disque du pool. La méthode recommandée de mise à jour des pools consiste à utiliser la commande <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> .

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 des pools de stockage ZFS” à la page 107
- “Visualisation des statistiques d'E/S des pools de stockage ZFS” à la page 111
- “Détermination de l'état de maintenance des pools de stockage ZFS” à la page 113

Affichage des informations des pools de stockage ZFS

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

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

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

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

La sortie de cette commande affiche les informations suivantes :

NAME	Nom du pool.
SIZE	Taille totale du pool, égale à la somme de la taille de tous les périphériques virtuels de niveau supérieur.
ALLOC	Quantité d'espace physique utilisée, c'est-à-dire allouée à tous les jeux de données et métadonnées internes. Notez que cette quantité d'espace disque est différente de celle qui est rapportée au niveau des systèmes de fichiers.

Pour de plus amples informations sur la détermination de l'espace de systèmes de fichiers disponible, reportez-vous à la section [“Comptabilisation de l'espace disque ZFS”](#) à la page 62.

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

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

HEALTH 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 113.

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 290.

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

```
# zpool list tank
NAME          SIZE    ALLOC  FREE   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 concernant tous les pools de stockage ou un pool spécifique”](#) à la page 107.

Script de sortie du pool de stockage ZFS

La sortie par défaut de la commande `zpool list` a été conçue pour améliorer la lisibilité. Elle n'est pas facile à utiliser en tant que partie d'un script shell. Pour faciliter l'utilisation de la commande dans le cadre de la programmation, l'option `-H` permet de supprimer les en-têtes de

colonnes et de séparer les champs par des onglets plutôt que par des espaces. La syntaxe suivante permet d'obtenir la liste des noms de pool du système :

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

Voici un autre exemple :

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

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

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

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

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

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

Les caractéristiques de l'historique sont les suivantes :

- Le journal ne peut pas être désactivé.
- Le journal est enregistré en permanence sur disque, c'est-à-dire qu'il est conservé d'une réinitialisation système à une autre.
- Le journal est implémenté en tant que tampon d'anneau. La taille minimale est de 128 Ko. La taille maximale est de 32 Mo.

- Pour des pools de taille inférieure, la taille maximale est plafonnée à 1 % de la taille du pool, la valeur *size* étant déterminée lors de la création du pool.
- Le journal ne nécessite aucune administration, ce qui signifie qu'il n'est pas nécessaire d'ajuster la taille du journal ou de modifier son emplacement.

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

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

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

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

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

```
# zpool history -i tank
2010-05-13.14:13:15 zpool create -f tank mirror c1t2d0 c1t3d0
2010-05-13.14:13:45 [internal pool create txg:6] pool spa 19; zfs spa 19; zpl 4;...
2010-05-13.14:21:19 zfs create tank/snaps
2010-05-13.14:22:02 [internal replay_inc_sync txg:20451] dataset = 41
2010-05-13.14:25:25 [internal snapshot txg:20480] dataset = 52
2010-05-13.14:25:25 [internal destroy_begin_sync txg:20481] dataset = 41
2010-05-13.14:25:26 [internal destroy txg:20488] dataset = 41
2010-05-13.14:25:26 [internal reservation set txg:20488] 0 dataset = 0
2010-05-14.08:10:29 zfs create tank/ws01
2010-05-14.08:10:54 [internal snapshot txg:53992] dataset = 42
2010-05-14.08:10:54 zfs snapshot tank/ws01@now
2010-05-14.08:11:04 [internal create txg:53994] dataset = 58
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix
```

Visualisation des statistiques d'E/S des 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 la commande `iostat`. Elle permet d'afficher un instantané statique de toutes les activités d'E/S, ainsi que les statistiques mises à jour pour chaque intervalle spécifié. Les statistiques suivantes sont rapportées :

<code>alloc capacity</code>	Capacité utilisée, c'est-à-dire quantité de données actuellement stockées dans le pool ou le périphérique. Cette quantité diffère quelque peu de la quantité d'espace disque disponible pour les systèmes de fichiers effectifs en raison de détails d'implémentation interne.
	Pour de plus amples informations sur la différence entre l'espace de pool et l'espace de jeux de données, reportez-vous à la section “Comptabilisation de l'espace disque ZFS” à la page 62.
<code>free capacity</code>	Capacité disponible, c'est-à-dire quantité d'espace disque disponible dans le pool ou le périphérique. Comme pour la capacité <code>used</code> , cette quantité diffère légèrement de la quantité d'espace disponible pour les jeux de données.
<code>read operations</code>	Nombre d'opérations de lecture d'E/S envoyées au pool ou au périphérique, y compris les requêtes de métadonnées.
<code>write operations</code>	Nombre d'opérations d'écriture d'E/S envoyées au pool ou au périphérique.
<code>read bandwidth</code>	Bande passante de toutes les opérations de lecture (métadonnées incluses), exprimée en unités par seconde.
<code>write bandwidth</code>	Bande passante de toutes les opérations d'écriture, exprimée en unités par seconde.

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

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

```
# zpool iostat
          capacity      operations      bandwidth
pool      alloc  free    read  write    read  write
-----
rpool     6.05G  61.9G     0     0     786   107
tank      31.3G  36.7G     4     1    296K  86.1K
-----
```

Comme ces statistiques sont cumulatives depuis 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      alloc  free    read  write    read  write
-----
tank      18.5G  49.5G      0    187      0  23.3M
tank      18.5G  49.5G      0    464      0  57.7M
tank      18.5G  49.5G      0    457      0  56.6M
tank      18.8G  49.2G      0    435      0  51.3M
```

Dans cet exemple, la commande affiche les statistiques d'utilisation pour le pool tank toutes les deux secondes, jusqu'à ce que vous saisissez Ctrl-C. Vous pouvez également spécifier un argument count supplémentaire pour entraîner l'interruption de la commande une fois le nombre spécifié d'itérations atteint. Par exemple, `zpool iostat 2 3` imprimerait un résumé toutes les deux secondes pour trois itérations, pendant six secondes. S'il n'y a qu'un pool unique, les statistiques s'affichent sur des lignes consécutives. S'il existe plusieurs pools, une ligne pointillée supplémentaire délimite chaque itération pour fournir une séparation visuelle.

Liste des statistiques d'E/S des périphériques virtuels

Outre les statistiques d'E/S à l'échelle du pool, la commande `zpool iostat` permet d'afficher des statistiques d'E/S pour des périphériques virtuels. Ainsi, vous pouvez identifier les périphériques anormalement lents ou consulter la répartition d'E/S générées par ZFS. Pour effectuer une 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
pool      alloc  free    read  write    read  write
-----
rpool      6.05G  61.9G      0      0      785   107
  mirror   6.05G  61.9G      0      0      785   107
    c1t0d0s0 -    -      0      0      578   109
    c1t1d0s0 -    -      0      0      595   109
-----
tank       36.5G  31.5G      4      1     295K  146K
  mirror   36.5G  31.5G     126    45    8.13M  4.01M
    c1t2d0 -    -      0      3     100K  386K
    c1t3d0 -    -      0      3     104K  386K
-----
```

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

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

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

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

ZFS offre une méthode intégrée pour examiner la maintenance des pools et des périphériques. La maintenance d'un pool se détermine par l'état de l'ensemble de ses périphériques. La commande `zpool status` permet d'afficher ces informations d'état. En outre, les défaillances potentielles des pools et des périphériques sont rapportées par la commande `fmtd`, s'affichent dans la console système et sont consignées dans le fichier `/var/adm/messages`.

Cette section décrit les méthodes permettant de déterminer la maintenance des pools et des périphériques. Ce chapitre n'aborde cependant pas les méthodes de réparation ou de récupération de pools en mauvais état de maintenance. Pour plus d'informations sur le dépannage et la récupération des données, reportez-vous au [Chapitre 11, "Dépannage d'Oracle Solaris ZFS et récupération de pool"](#).

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

ONLINE	Le périphérique ou le périphérique virtuel fonctionne normalement. Même si certaines erreurs transitoires peuvent encore survenir, le périphérique fonctionne correctement.
DEGRADED	Le périphérique virtuel a connu un panne. Toutefois, il continue de fonctionner. Cet état est le plus commun lorsqu'un miroir ou un périphérique RAID-Z a perdu un ou plusieurs périphériques le constituant. La tolérance de pannes du pool peut être compromise dans la mesure où une défaillance ultérieure d'un autre périphérique peut être impossible à résoudre.
FAULTED	Le périphérique ou 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 a été mis hors ligne explicitement par l'administrateur.
UNAVAIL	L'ouverture du périphérique ou du périphérique virtuel est impossible. Dans certains cas, les pools avec des périphériques en état UNAVAIL s'affichent en mode DEGRADED. Si un périphérique de niveau supérieur est en état UNAVAIL, aucun élément du pool n'est accessible.
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 UNAVAIL, 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 renvoyant l'état FAULTED est totalement inaccessible. Aucune donnée ne peut être récupérée tant que les périphériques nécessaires n'ont pas été connectés ou réparés. Un pool renvoyant l'état DEGRADED continue à être exécuté. Cependant, il se peut que vous ne puissiez pas atteindre le même niveau de redondance ou de capacité de données que s'il se trouvait en ligne.

État de maintenance de base de pool de stockage

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

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

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

État de maintenance détaillé

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

```
# zpool status -v tank
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
```

```
scrub: scrub completed after 0h0m with 0 errors on Wed Jan 20 15:13:59 2010
config:
```

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

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

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

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

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

```
# zpool status -x
pool: tank
state: DEGRADED
status: One or more devices has been taken offline by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 15:15:09 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM	
tank	DEGRADED	0	0	0	
mirror-0	DEGRADED	0	0	0	
c1t0d0	ONLINE	0	0	0	
c1t1d0	OFFLINE	0	0	0	48K resilvered

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

Les colonnes `READ` et `WRITE` indiquent le nombre d'erreurs d'E/S détectées dans le périphérique, tandis que la colonne `CKSUM` indique le nombre d'erreurs de somme de contrôle impossible à

corriger qui se sont produites sur le périphérique. Ces deux comptes d'erreurs indiquent une défaillance potentielle du périphérique et que des actions correctives sont requises. Si le nombre d'erreurs est non nul pour un périphérique virtuel de niveau supérieur, il est possible que des parties de vos données soient inaccessibles.

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

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

Pour plus d'informations sur le diagnostic et la réparation de pools et de données défaillants, reportez-vous au [Chapitre 11, “Dépannage d'Oracle Solaris ZFS et récupération de pool”](#).

Migration de pools de stockage ZFS

Parfois, il est possible que vous deviez déplacer un pool de stockage d'un système à l'autre. Pour ce faire, les périphériques de stockage doivent être déconnectés du système d'origine et reconnectés au système de destination. Pour accomplir cette tâche, vous pouvez raccorder physiquement les périphériques ou utiliser des périphériques multiport, par exemple les périphériques d'un SAN. Le système de fichiers ZFS permet d'exporter le pool d'une machine et de l'importer sur le système de destination, même si l'endianisme de l'architecture des systèmes est différente. Pour plus d'informations sur la réplication ou la migration de systèmes de fichiers d'un pool de stockage à un autre résidant éventuellement sur des machines différentes, reportez-vous à la section [“Envoi et réception de données ZFS”](#) à la page 234.

- [“Préparatifs de migration de pool de stockage ZFS”](#) à la page 116
- [“Exportation d'un pool de stockage ZFS”](#) à la page 117
- [“Définition des pools de stockage disponibles pour importation”](#) à la page 117
- [“Importation de pools de stockage ZFS à partir d'autres répertoires”](#) à la page 119
- [“Importation de pools de stockage ZFS”](#) à la page 120
- [“Récupération de pools de stockage ZFS détruits”](#) à la page 121

Préparatifs de migration de pool de stockage ZFS

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

Si vous retirez les disques manuellement, au lieu d'exporter le pool explicitement, vous pouvez toujours importer le pool résultant dans un autre système. Cependant, vous pourriez perdre les dernières secondes de transactions de données et le pool s'affichera alors comme étant défaillant sur le système d'origine dans la mesure où les périphériques ne sont plus présents. Par défaut, le

système de destination refuse d'importer un pool qui n'a pas été exporté implicitement. Cette condition est nécessaire car elle évite les importations accidentelles d'un pool actif composé de stockage connecté au réseau toujours en cours d'utilisation sur un autre système.

Exportation d'un pool de stockage ZFS

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

```
# zpool export tank
```

La commande tente de démonter tout système de fichiers démonté au sein du pool avant de continuer. Si le démontage d'un des 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
```

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

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

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

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

Définition des pools de stockage disponibles pour importation

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

```
# zpool import
pool: tank
  id: 11809215114195894163
  state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

    tank          ONLINE
    mirror-0     ONLINE
    c1t0d0       ONLINE
    c1t1d0       ONLINE
```

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

Tout comme la sortie de la commande `zpool status`, la sortie de la commande `zpool import` se rapporte à un article de connaissances contenant les informations les plus récentes sur les procédures de réparation pour les problèmes qui empêchent l'importation d'un pool. Dans ce cas, l'utilisateur peut forcer l'importation du pool. Cependant, l'importation d'un pool en cours d'utilisation par un autre système au sein d'un réseau de stockage peut entraîner une 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 des données redondantes suffisantes sont disponibles pour obtenir un pool utilisable, le pool s'affiche dans l'état DEGRADED. Exemple :

```
# zpool import
pool: tank
  id: 11809215114195894163
  state: DEGRADED
status: One or more devices are missing from the system.
action: The pool can be imported despite missing or damaged devices. The
       fault tolerance of the pool may be compromised if imported.
  see: http://www.sun.com/msg/ZFS-8000-2Q
config:

    NAME          STATE          READ WRITE CKSUM
    tank          DEGRADED          0    0    0
    mirror-0     DEGRADED          0    0    0
    c1t0d0       UNAVAIL          0    0    0  cannot open
    c1t3d0       ONLINE            0    0    0
```

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

```
# zpool import
pool: dozer
  id: 9784486589352144634
  state: FAULTED
action: The pool cannot be imported. Attach the missing
```

```

devices and try again.
see: http://www.sun.com/msg/ZFS-8000-6X
config:
raidz1-0      FAULTED
c1t0d0       ONLINE
c1t1d0       FAULTED
c1t2d0       ONLINE
c1t3d0       FAULTED

```

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

```

# zpool import
pool: dozer
  id: 9784486589352144634
  state: FAULTED
status: One or more devices are missing from the system.
action: The pool cannot be imported. Attach the missing
        devices and try again.
  see: http://www.sun.com/msg/ZFS-8000-6X
config:
dozer      FAULTED  missing device
raidz1-0   ONLINE
c1t0d0     ONLINE
c1t1d0     ONLINE
c1t2d0     ONLINE
c1t3d0     ONLINE

```

Additional devices are known to be part of this pool, though their exact configuration cannot be determined.

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

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

```

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

dozer      ONLINE

```

```

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

```

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

Importation de pools de stockage ZFS

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

```
# zpool import tank
```

Si plusieurs pools disponibles possèdent le même nom, vous devez spécifier le pool à importer à l'aide de l'identifiant numérique. 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 290](#).

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: 5154272182900538157
  state: ONLINE (DESTROYED)
 action: The pool can be imported using its name or numeric identifier.
config:

      tank          ONLINE
      mirror-0      ONLINE
      c1t0d0         ONLINE
      c1t1d0         ONLINE
```

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

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

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

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

      NAME          STATE          READ WRITE CKSUM
      tank          ONLINE
      mirror-0      ONLINE
      c1t0d0         ONLINE
      c1t1d0         ONLINE

errors: No known data errors
```

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

```
# zpool destroy dozer
# zpool import -D
pool: dozer
  id: 13643595538644303788
  state: DEGRADED (DESTROYED)
  status: One or more devices could not be opened. Sufficient replicas exist for
         the pool to continue functioning in a degraded state.
  action: Attach the missing device and online it using 'zpool online'.
  see: http://www.sun.com/msg/ZFS-8000-2Q
  config:
```

NAME	STATE	READ	WRITE	CKSUM	
dozer	DEGRADED	0	0	0	
raidz2-0	DEGRADED	0	0	0	
c2t8d0	ONLINE	0	0	0	
c2t9d0	ONLINE	0	0	0	
c2t10d0	ONLINE	0	0	0	
c2t11d0	UNAVAIL	0	35	1	cannot open
c2t12d0	ONLINE	0	0	0	

```
errors: No known data errors
# zpool import -Df dozer
# zpool status -x
pool: dozer
  state: DEGRADED
  status: One or more devices could not be opened. Sufficient replicas exist for
         the pool to continue functioning in a degraded state.
  action: Attach the missing device and online it using 'zpool online'.
  see: http://www.sun.com/msg/ZFS-8000-2Q
  scrub: scrub completed after 0h0m with 0 errors on Thu Jan 21 15:38:48 2010
  config:
```

NAME	STATE	READ	WRITE	CKSUM	
dozer	DEGRADED	0	0	0	
raidz2-0	DEGRADED	0	0	0	
c2t8d0	ONLINE	0	0	0	
c2t9d0	ONLINE	0	0	0	
c2t10d0	ONLINE	0	0	0	
c2t11d0	UNAVAIL	0	37	0	cannot open
c2t12d0	ONLINE	0	0	0	

```
errors: No known data errors
# zpool online dozer c2t11d0
Bringing device c2t11d0 online
# zpool status -x
all pools are healthy
```

Mise à niveau de pools de stockage ZFS

Si certains pools de stockage ZFS proviennent d'une version antérieure de Solaris (la version Solaris 10 10/09, 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 actuelle. De plus, la commande `zpool status` indique dorénavant si la version actuelle des pools est plus ancienne. Exemple :

```
# zpool status
pool: tank
state: ONLINE
status: The pool is formatted using an older on-disk format. The pool can
still be used, but some features are unavailable.
action: Upgrade the pool using 'zpool upgrade'. Once this is done, the
pool will no longer be accessible on older software versions.
scrub: none requested
config:
    NAME          STATE          READ WRITE CKSUM
    tank          ONLINE         0     0     0
    mirror-0     ONLINE         0     0     0
    c1t0d0       ONLINE         0     0     0
    c1t1d0       ONLINE         0     0     0
errors: No known data errors
```

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

```
# zpool upgrade -v
This system is currently running ZFS pool version 22.
```

The following versions are supported:

```
VER  DESCRIPTION
-----
 1  Initial ZFS version
 2  Ditto blocks (replicated metadata)
 3  Hot spares and double parity RAID-Z
 4  zpool history
 5  Compression using the gzip algorithm
 6  bootfs pool property
 7  Separate intent log devices
 8  Delegated administration
 9  refquota and reservation properties
10  Cache devices
11  Improved scrub performance
12  Snapshot properties
13  snapused property
14  passthrough-x aclinherit
15  user/group space accounting
16  stmf property support
17  Triple-parity RAID-Z
18  Snapshot user holds
19  Log device removal
20  Compression using zle (zero-length encoding)
```

- 21 Reserved
- 22 Received properties

For more information on a particular version, including supported releases, see the ZFS Administration Guide.

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

```
# zpool upgrade -a
```

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

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

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

Il contient les sections suivantes :

- “Installation et initialisation d'un système de fichiers racine Oracle Solaris ZFS (présentation)” à la page 126
- “Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS” à la page 127
- “Installation d'un système de fichiers racine ZFS (installation initiale)” à la page 130
- “Installation d'un système de fichiers racine ZFS (installation d'archive Oracle Solaris Flash)” à la page 138
- “Installation d'un système de fichiers racine ZFS (installation d'Oracle Solaris JumpStart)” à la page 140
- “Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Oracle Solaris Live Upgrade)” à la page 144
- “Prise en charge ZFS des périphériques de swap et de vidage” à la page 167
- “Initialisation à partir d'un système de fichiers racine ZFS” à la page 171
- “Restauration du pool racine ZFS ou des instantanés du pool racine” à la page 178

Vous trouverez la liste des problèmes connus relatifs à cette version dans les *Notes de version Oracle Solaris 10 9/10*.

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 Oracle Solaris ZFS (présentation)

À partir de la version Solaris 10 10/08, vous pouvez installer et initialiser un système de fichiers racine ZFS des manières suivantes :

- Il est possible d'effectuer une installation initiale lorsque ZFS est sélectionné en tant que système de fichiers racine.
- Vous pouvez utiliser la fonction Oracle Solaris Live Upgrade pour migrer d'un système de fichiers racine UFS vers un système de fichiers racine ZFS. Vous pouvez également utiliser Oracle 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.
- Vous pouvez utiliser un profil Oracle Solaris JumpStart pour installer automatiquement un système avec un système de fichiers racine ZFS.
- À partir de la version Solaris 10 10/09, vous pouvez utiliser un profil JumpStart pour installer automatiquement un système avec une archive Flash ZFS.

Une fois qu'un système SPARC ou x86 est installé avec un système de fichiers 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 171.

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 :
 - SPARC : utilisez la syntaxe suivante à partir du DVD d'installation de Solaris :
`ok boot cdrom - text`
 - SPARC : utilisez la syntaxe suivante lors d'une initialisation à partir du réseau :
`ok boot net - text`
 - x86 : sélectionnez l'option d'installation en mode texte.
- Un profil JumpStart personnalisé permet d'accéder aux fonctionnalités suivantes :
 - Vous pouvez définir un profil pour la création d'un pool de stockage ZFS et désigner un système de fichiers ZFS d'initialisation.

- Vous pouvez définir un profil pour identifier une archive Flash d'un pool racine ZFS.
- Vous pouvez utiliser la fonction Oracle 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.
- 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 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 Oracle Solaris Flash pour l'installation d'un système de fichiers racine ZFS n'est pas disponible en sélectionnant l'option d'installation Flash à partir l'option d'installation initiale. Cependant, vous pouvez créer un profil JumpStart pour identifier une archive Flash d'un pool racine ZFS. Pour plus d'informations, reportez-vous à la section [“Installation d'un système de fichiers racine ZFS \(installation d'archive Oracle Solaris Flash\)”](#) à la page 138.
- 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.

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

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

Version Oracle Solaris requise

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

- Installation d'un système de fichiers racine ZFS - disponible à partir de la version 10 10/08 de Solaris.

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

Configuration requise générale relative au pool de stockage ZFS

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

Espace disque requis pour les pools 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, le périphériques de swap et le 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.

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

- 768 Mo de mémoire pour installer un système de fichiers ZFS racine ;
- 1 Go de mémoire pour de meilleures performances globales ;
- 16 Go d'espace disque minimum recommandés. L'espace disque est utilisé comme suit :
 - **Zone de swap et périphérique de vidage** : les tailles par défaut des volumes de swap et de vidage créés par le programme d'installation de Solaris sont les suivantes :
 - **Installation initiale de Solaris** : dans le nouvel environnement d'initialisation ZFS, la taille du volume de swap par défaut est calculée selon la moitié de la taille de la mémoire physique, de 512 Mo à 2 Go en règle générale. Vous pouvez ajuster la taille du volume de swap au cours de l'installation initiale.
 - La taille par défaut du volume de vidage est calculée par le noyau en fonction des informations dumpadm et de la taille de la mémoire physique. Vous pouvez régler la taille du volume de vidage au cours de l'installation initiale.
 - **Oracle Solaris Live Upgrade** : lors de la migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS, la taille par défaut du volume de swap de l'environnement d'initialisation ZFS est calculée selon la taille du périphérique de swap de l'environnement d'initialisation UFS. La taille par défaut du volume de swap est calculée en additionnant la taille de tous les périphériques de swap de l'environnement d'initialisation UFS. Un volume ZFS de cette taille est ensuite créé

dans l'environnement d'initialisation ZFS. Si aucun périphérique de swap n'est défini dans l'environnement d'initialisation UFS, la taille par défaut du volume d'échange est définie sur 512 Mo.

- Dans l'environnement d'initialisation ZFS, la taille du volume de vidage par défaut est définie selon la moitié de la taille de la mémoire physique. Cela va généralement de 512 Mo à 2 Go.

Vous pouvez ajuster la taille de vos volumes de swap et de vidage, dès lors que ces dernières prennent en charge les opérations du système. Pour plus d'informations, reportez-vous à la section [“Ajustement de la taille de vos périphériques de swap et de vidage ZFS” à la page 168.](#)

- **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 nécessite environ 6 Go. Chaque environnement d'initialisation ZFS cloné à partir d'un autre environnement d'initialisation ZFS ne nécessite aucun espace disque supplémentaire ; toutefois, lors de l'application de patches, la taille de l'environnement d'initialisation peut augmenter. Tous les environnements d'initialisation ZFS d'un même pool racine utilisent les mêmes périphériques de swap et de vidage.
- **Composants du SE Solaris** : tous les sous-répertoires du système de fichiers racine faisant partie de l'image du SE doivent se trouver dans le même jeu de données en tant que système de fichiers racine, à l'exception du fichier `/var`. En outre, tous les composants du SE Solaris doivent se trouver dans le pool racine, à l'exception des périphériques de swap et de vidage.

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

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

Configuration requise relative au pool de stockage ZFS

Vérifiez la configuration requise suivante pour le pool de stockage ZFS :

- Le pool utilisé comme pool racine doit contenir une étiquette SMI. Cette condition doit être respectée si le pool est créé avec des tranches de disque.

- Le pool doit exister sur une tranche de disque ou sur des tranches de disque qui sont mises en miroir. Si vous tentez d'utiliser une configuration de pool non prise en charge lors d'une migration effectuée par Oracle Live Upgrade, un message semblable au message suivant s'affiche :

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

Pour obtenir une description détaillée des configurations de pool racine ZFS prises en charge, reportez-vous à la section [“Création d'un pool racine ZFS”](#) à la page 73.

- x86 : le disque doit comporter une partition Solaris `fdisk`. Une partition `fdisk` Solaris est créée automatiquement lors de l'installation du système x86. Pour plus d'informations sur les partitions `fdisk` Solaris, reportez-vous à la section [“Guidelines for Creating an fdisk Partition”](#) du *System Administration Guide: Devices and File Systems*.
- 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.
- La compression peut être activée sur le pool racine uniquement après l'installation de ce dernier. Aucun moyen ne permet d'activer la compression sur un pool racine au cours de l'installation. L'algorithme de compression `gzip` n'est pas pris en charge sur les pools racine.
- Ne renommez pas le pool racine après sa création par une installation initiale ou une fois la migration d'Oracle Solaris Live Upgrade vers un système de fichiers racine ZFS effectuée. Si vous renommez le pool racine, cela peut empêcher l'initialisation du système.

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 à utiliser pour votre système de fichiers racine ZFS, vous devez utiliser

Oracle 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 \(Oracle Solaris Live Upgrade\)”](#) à la page 144.

Si vous décidez de configurer des zones après l'installation initiale d'un système de fichiers racine ZFS et si vous prévoyez d'appliquer un patch au système ou de le mettre à niveau, reportez-vous aux sections [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones \(Solaris 10 10/08\)”](#) à la page 151 ou [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones \(version Solaris 5 10/09 ou supérieure\)”](#) à la page 156.

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

There are existing ZFS pools available on this system. However, they can only be upgraded using the Live Upgrade tools. The following screens will only allow you to install a ZFS root system, not upgrade one.



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

Avant de lancer l'installation initiale pour créer un pool de stockage ZFS, reportez-vous à la section [“Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS”](#) à la page 127.

EXEMPLE 5-1 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 versions précédentes 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. Les étapes à suivre pour installer un système de fichiers racine ZFS sont les suivantes :

1. Sélectionnez la méthode d'installation interactive de Solaris car aucune installation Solaris Flash n'est disponible pour créer un système de fichiers racine ZFS d'initialisation. Cependant, vous pouvez créer une archive Flash ZFS à utiliser pendant une installation JumpStart. Pour plus d'informations, reportez-vous à la section [“Installation d'un système de fichiers racine ZFS \(installation d'archive Oracle Solaris Flash\)”](#) à la page 138.

À compter de la version Solaris 10 10/08, vous pouvez migrer un système de fichier racine UFS vers un système de fichiers racine ZFS, à condition que la version Solaris 10 10/08 ou une version ultérieure soit déjà installée. Pour plus d'informations sur la migration vers un système de fichiers racine ZFS, reportez-vous à la section [“Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS \(Oracle Solaris Live Upgrade\)”](#) à la page 144.

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 vous avez sélectionné le logiciel à installer, vous êtes invité à sélectionner les disques pour créer votre pool de stockage ZFS. Cet écran est similaire à celui des versions précédentes de Solaris.

EXEMPLE 5-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

Select Disks

On this screen you must select the disks for installing Solaris software. Start by looking at the Suggested Minimum field; this value is the approximate space needed to install the software you've selected. For ZFS, multiple disks will be configured as mirrors, so the disk you choose, or the slice within the disk must exceed the Suggested Minimum value.

NOTE: ** denotes current boot disk

Disk Device	Available Space
[X] c1t0d0	69994 MB (F4 to edit)
[] c1t1d0	69994 MB
[-] c1t2d0	0 MB
[-] c1t3d0	0 MB

Maximum Root Size: 69994 MB

Suggested Minimum: 8279 MB

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 comportant 2 ou 3 disques est optimal. Si vous disposez de 8 disques et que vous les sélectionnez tous, ces 8 disques seront utilisés pour le pool racine comme miroir géant. Cette configuration n'est pas optimale. Une autre possibilité consiste à créer un pool racine mis en miroir une fois l'installation initiale 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 69.

- Pour sélectionner 2 disques afin de créer un pool racine mis en miroir, utilisez les touches fléchées pour sélectionner le deuxième disque. Dans l'exemple suivant, les deux disques c1t1d0 et c1t2d0 sont sélectionnés en tant que disques de pool racine. Les deux disques doivent posséder une étiquette SMI et une tranche 0. Si les disques ne sont pas identifiés par une étiquette SMI ou ne contiennent aucune tranche, vous devez quitter le programme d'installation, utiliser l'utilitaire de formatage pour réétiqueter et repartitionner les disques, puis relancer le programme d'installation.

Select Disks

On this screen you must select the disks for installing Solaris software. Start by looking at the Suggested Minimum field; this value is the approximate space needed to install the software you've selected. For ZFS, multiple disks will be configured as mirrors, so the disk you choose, or the slice within the disk must exceed the Suggested Minimum value.

NOTE: ** denotes current boot disk

Disk Device	Available Space
[X] c1t0d0	69994 MB
[X] c1t1d0	69994 MB (F4 to edit)
[-] c1t2d0	0 MB
[-] c1t3d0	0 MB

EXEMPLE 5-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

```
Maximum Root Size: 69994 MB
Suggested Minimum: 8279 MB
```

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

Si vous ne créez pas de pool racine mis en miroir lors de l'installation, vous pouvez facilement en créer un après l'installation. Pour plus d'informations, reportez-vous à la section [“Création d'un pool racine mis en miroir \(post-installation\)”](#) à la page 136.

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

Configure ZFS Settings

```
Specify the name of the pool to be created from the disk(s) you have chosen.
Also specify the name of the dataset to be created within the pool that is
to be used as the root directory for the filesystem.
```

```
          ZFS Pool Name: rpool
    ZFS Root Dataset Name: s10s_u9wos_08
    ZFS Pool Size (in MB): 69995
    Size of Swap Area (in MB): 2048
    Size of Dump Area (in MB): 1536
    (Pool size must be between 6231 MB and 69995 MB)

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

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 de jeu de données racine est remplacé par `zfsBE`.

```
          ZFS Pool Name: rpool
    ZFS Root Dataset Name: zfsBE
    ZFS Pool Size (in MB): 69995
    Size of Swap Area (in MB): 2048
    Size of Dump Area (in MB): 1536
    (Pool size must be between 6231 MB and 69995 MB)

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

6. Vous pouvez modifier le profil d'installation dans ce dernier écran de l'installation. Exemple :

EXEMPLE 5-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

Profile

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

```

=====
Installation Option: Initial
      Boot Device: c1t0d0
Root File System Type: ZFS
      Client Services: None

      Regions: North America
      System Locale: C ( C )

      Software: Solaris 10, Entire Distribution
      Pool Name: rpool
Boot Environment Name: zfsBE
      Pool Size: 69995 MB
      Devices in Pool: c1t0d0
                      c1t1d0

```

7. Une fois l'installation terminée, vérifiez les informations du pool de stockage et du 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
  mirror-0    ONLINE    0     0     0
    c1t0d0s0  ONLINE    0     0     0
    c1t1d0s0  ONLINE    0     0     0

errors: No known data errors
# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
rpool         8.03G 58.9G   96K    /rpool
rpool/ROOT    4.47G 58.9G   21K    legacy
rpool/ROOT/zfsBE 4.47G 58.9G  4.47G    /
rpool/dump    1.50G 58.9G  1.50G    -
rpool/export  44K   58.9G  23K    /export
rpool/export/home 21K   58.9G  21K    /export/home
rpool/swap    2.06G 61.0G  16K    -

```

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

8. 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 `zfs2BE` est créé. L'environnement d'initialisation actuel s'appelle

EXEMPLE 5-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

zfsBE, comme l'indique la sortie `zfs list`. Toutefois, tant que le nouvel environnement d'initialisation n'est pas créé, l'environnement d'initialisation actuel n'est pas reconnu par la sortie `lustatus`.

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

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

```
# lucreate -n zfs2BE
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

La création d'un environnement d'exécution ZFS au sein du même pool utilise les fonctions de clonage et d'instantané ZFS pour créer instantanément l'environnement d'exécution. Pour plus d'informations sur l'utilisation d'Oracle 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 \(Oracle Solaris Live Upgrade\)”](#) à la page 144.

9. Vérifiez ensuite les nouveaux environnements d'initialisation. Exemple :

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
zfsBE                  yes     yes   yes     no    -
zfs2BE                 yes     no    no      yes   -

# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                8.03G 58.9G   97K   /rpool
rpool/ROOT           4.47G 58.9G   21K   legacy
rpool/ROOT/zfs2BE    116K  58.9G  4.47G   /
rpool/ROOT/zfsBE     4.47G 58.9G  4.47G   /
rpool/ROOT/zfsBE@zfs2BE 75.5K -      4.47G   -
rpool/dump           1.50G 58.9G  1.50G   -
```

EXEMPLE 5-1 Installation initiale d'un système de fichiers racine ZFS d'initialisation (Suite)

```

rpool/export          44K  58.9G   23K  /export
rpool/export/home    21K  58.9G   21K  /export/home
rpool/swap           2.06G  61.0G   16K  -

```

10. Pour initialiser un système à partir d'un autre environnement d'initialisation, 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 initialiser le système à partir du nouvel environnement d'initialisation `zfs2BE`, sélectionnez l'option 2. Saisissez ensuite la commande `boot -Z` affichée.

```

ok boot -L
Executing last command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0 File and args: -L
1 zfsBE
2 zfs2BE
Select environment to boot: [ 1 - 2 ]: 2

To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfs2BE
ok boot -Z rpool/ROOT/zfs2BE

```

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 171.

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

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

Pour obtenir des informations sur le remplacement d'un disque dans le pool racine ZFS, reportez-vous à la section [“Remplacement d'un disque dans le pool racine ZFS”](#) à la page 178.

1 Affichez l'état du pool racine actuel.

```

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

        NAME        STATE        READ WRITE CKSUM
        rpool        ONLINE        0     0     0

```



```
c1t0d0s0 ONLINE      0      0      0
```

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

2 Connectez un deuxième disque pour configurer un pool racine mis en miroir.

```
# zpool attach rpool c1t0d0s0 c1t1d0s0
```

Please be sure to invoke `installboot(1M)` to make 'c1t1d0s0' bootable.
Make sure to wait until `resilver` is done before rebooting.

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

```
# zpool status rpool
```

```
pool: rpool
state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
        scrub: resilver in progress for 0h1m, 24.26% done, 0h3m to go
config:
```

NAME	STATE	READ	WRITE	CKSUM	
rpool	ONLINE	0	0	0	
mirror-0	ONLINE	0	0	0	
c1t0d0s0	ONLINE	0	0	0	
c1t1d0s0	ONLINE	0	0	0	3.18G resilvered

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

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

```
scrub: resilver completed after 0h10m with 0 errors on Thu Mar 11 11:27:22 2010
```

4 Appliquez les blocs d'initialisation au deuxième disque, une fois la réargenture terminée.

```
sparc# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t1d0s0
```

```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

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

6 Configurez le système pour qu'il puisse s'initialiser automatiquement à partir du nouveau disque à l'aide de la commande `eeprom` ou de la commande `setenv` de la PROM d'initialisation SPARC. Vous pouvez également reconfigurer le BIOS du PC.

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

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

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

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

- Seule l'installation JumpStart d'une archive Flash ZFS est prise en charge. Vous ne pouvez pas utiliser l'option d'installation interactive d'une archive Flash pour installer un système avec un système de fichiers racine ZFS. Vous ne pouvez pas non plus utiliser une archive Flash pour installer un environnement d'initialisation ZFS avec Oracle Solaris Live Upgrade.
- Vous pouvez uniquement installer une archive Flash sur un système doté de la même architecture que le système sur lequel vous avez créé l'archive Flash ZFS. Par exemple, une archive créée sur un système `sun4u` ne peut pas être installée sur un système `sun4v`.
- Seule une nouvelle installation complète d'une archive Flash ZFS est prise en charge. Vous ne pouvez pas installer d'archive Flash différentielle d'un système de fichiers racine ZFS, ni installer une archive UFS/ZFS hybride.
- Les archives Flash UFS existantes peuvent toujours être utilisées, mais uniquement pour installer un système de fichiers racine UFS. L'archive Flash ZFS peut uniquement être utilisée pour installer un système de fichiers racine ZFS.
- Bien que la totalité du pool racine, mis à part les jeux de données exclus explicitement, soit archivée et installée, seul l'environnement d'initialisation ZFS qui est initialisé lors de la création de l'archive est utilisable après l'installation de l'archive Flash. Cependant, les pools archivés avec l'option `R rootdir` de la commande `-flar` ou `flarcreate` peuvent être utilisés pour archiver un autre pool racine que celui en cours d'initialisation.
- Un nom de pool racine ZFS créé avec une archive Flash doit correspondre au nom du pool racine principal. Le nom du pool racine utilisé pour créer l'archive Flash correspond au nom attribué au pool créé. La modification du nom du pool n'est pas prise en charge.

- Les options des commandes `flarcreate` et `flar` d'inclusion et d'exclusion des fichiers individuels ne sont pas prises en charge dans une archive Flash ZFS. Vous pouvez uniquement exclure des jeux de données entiers à partir d'une archive Flash ZFS.
- La commande `flar info` n'est pas prise en charge par une archive Flash ZFS. Exemple :

```
# flar info -l zfs10u8flar
ERROR: archive content listing not supported for zfs archives.
```

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

- Installez la version Solaris 10 10/09 ou mettez votre version à niveau sur le système principal. Ajoutez toutes les personnalisations que vous souhaitez.
- Créez l'archive Flash ZFS avec la commande `flarcreate` sur le système principal. Tous les jeux de données du pool racine sont inclus dans l'archive Flash ZFS, à l'exception des volumes d'échange et de vidage.
- Créez un profil JumpStart pour inclure les informations d'archive Flash sur le serveur d'installation.
- Installez l'archive Flash ZFS sur le système cible.

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

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

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

- L'ensemble de la hiérarchie du jeu de données qui existait sur le système sur lequel l'archive Flash a été créée est recréé sur le système cible, mis à part tous les jeux de données qui ont été spécifiquement exclus au moment de la création de l'archive. Les volumes de swap et de vidage ne sont pas inclus dans l'archive Flash.
- Le pool racine possède le même nom que le pool qui a été utilisé pour créer l'archive.
- L'environnement d'initialisation actif au moment où l'archive Flash a été créée correspond à l'environnement d'initialisation actif et par défaut sur les systèmes déployés.

EXEMPLE 5-2 Installation d'un système avec une archive Flash ZFS

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

EXEMPLE 5-2 Installation d'un système avec une archive Flash ZFS (Suite)

```
# flarcreate -n zfsBE zfs10upflar
Full Flash
Checking integrity...
Integrity OK.
Running precreation scripts...
Precreation scripts done.
Determining the size of the archive...
The archive will be approximately 4.94GB.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.

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

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

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

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

Vous pouvez créer un profil JumpStart pour installer un système de fichiers racine ZFS ou UFS.

Un profil spécifique au système de fichiers ZFS doit contenir le nouveau mot-clé `pool`. Le mot-clé `pool` installe un nouveau pool racine et un nouvel environnement d'initialisation est créé par défaut. Vous pouvez fournir le nom de l'environnement d'initialisation et créer un jeu de données `/var` distinct à l'aide des mots-clés `bootenv` `installbe` et des options `bename` et `dataset`.

Pour plus d'informations sur l'utilisation des fonctions JumpStart, reportez-vous au [Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée](#).

Si vous décidez de configurer des zones après l'installation initiale d'un système de fichiers racine ZFS et si vous prévoyez l'application d'un patch au système ou sa mise à niveau, reportez-vous aux sections “[Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones \(Solaris 10 10/08\)](#)” à la page 151 ou “[Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones \(version Solaris 5 10/09 ou supérieure\)](#)” à la page 156.

Mots-clés JumpStart pour ZFS

Les mots-clés suivants sont autorisés dans un profil spécifique au système de fichiers ZFS :

auto Spécifie automatiquement la taille des tranches du pool, du volume d'échange ou de vidage. La taille du disque est contrôlée pour s'assurer que la taille minimale peut être satisfaite. Plus la taille est réduite, plus la taille du pool allouée est grande. Cette taille dépend de certaines contraintes, notamment de la taille des disques, des tranches conservées, etc.

Par exemple, lorsque vous spécifiez `cot0d0s0`, une tranche de pool racine de taille optimale est créée si vous spécifiez les mots-clés `all` ou `auto`. Vous pouvez également indiquer une taille de tranche, de volume de swap ou de vidage spécifique.

Le mot-clé `auto` fonctionne de la même façon que le mot-clé `all`, lorsqu'il est utilisé avec un pool racine ZFS car les pools ne disposent pas d'espace inutilisé.

bootenv Identifie les caractéristiques de l'environnement d'initialisation.

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 :

```
pool 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*). La valeur *poolname* ne doit pas identifier le nom d'un pool existant car celui-ci serait écrasé.

poolsize Spécifie la taille du pool à créer. La valeur peut être `auto` ou `existing`. La valeur `auto` attribue une taille de pool optimale 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 non pas écrasées. L'unité de taille supposée s'exprime en Mo, à moins de spécifier l'option `g` (Go).

swapsize Spécifie la taille du volume d'échange à créer. La valeur `auto` signifie que la taille de swap par défaut est utilisée. Vous pouvez spécifier une taille avec une valeur *size*. L'unité de taille supposée s'exprime en Mo, à moins de spécifier l'option `g` (Go).

dumpsize Spécifie la taille du volume de vidage à créer. La valeur `auto` signifie que la taille de swap par défaut est utilisée. Vous pouvez spécifier une taille avec une valeur *size*. L'unité de taille supposée s'exprime en Mo, à moins de spécifier `g` (Go).

vdevlist Spécifie un ou plusieurs périphériques à utiliser pour créer le pool. Le format de l'option *vdevlist* est identique au format de la commande `zpool create`. À l'heure actuelle, seules les configurations mises en miroir sont prises en charge lorsque plusieurs périphériques sont spécifiés. Les périphériques de l'option *vdevlist* doivent correspondre à des tranches du pool racine. La valeur `any` signifie que le logiciel d'installation sélectionne le périphérique approprié.

Vous pouvez mettre en miroir autant de disques que vous le souhaitez, mais la taille du pool créé est déterminée par le plus petit des disques spécifiés. Pour plus d'informations sur la création de pools de stockage mis en miroir, reportez-vous à la section [“Configuration de pool de stockage mis en miroir”](#) à la page 69.

Exemples de profils JumpStart pour ZFS

Cette section fournit des exemples de profils JumpStart spécifiques à ZFS.

Le profil suivant effectue une installation initiale spécifiée avec `install_type initial-install` dans un nouveau pool, identifié par pool `newpool`, dont la taille est automatiquement définie sur la taille des disques spécifiés par le mot-clé `auto`. La zone de swap et le périphérique de vidage sont automatiquement dimensionnés par le mot-clé `auto` dans une configuration de disques mis en miroir (mot-clé `mirror` et disques `c0t0d0s0` et `c0t1d0s0`). Les caractéristiques de l'environnement d'initialisation sont définies par le mot-clé `bootenv` afin d'installer un nouvel environnement d'initialisation avec le mot-clé `installbe` ; un environnement (bename) portant le nom `s10-xx` est créé.

```
install_type initial_install
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename s10-xx
```

Le profil suivant effectue une installation initiale avec le mot-clé `install_type initial-install` du méta-cluster `SUNWCall` dans un nouveau pool de 80 Go portant le nom `newpool`. Ce pool est créé avec un volume d'échange et un volume de vidage de 2 Go chacun dans une configuration mise en miroir dans l'un des deux périphériques disponibles dont la taille permet de créer un pool de 80 Go. Si deux périphériques de ce type ne sont pas disponibles, l'installation échoue. Les caractéristiques de l'environnement d'initialisation sont définies par le mot-clé `bootenv` afin d'installer un nouvel environnement d'initialisation avec le mot-clé `installbe` ; un environnement (bename) portant le nom `s10-xx` est créé.

```
install_type initial_install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename s10-xx
```

La syntaxe d'installation JumpStart permet de conserver ou de créer un système de fichiers UFS sur un disque contenant également un pool racine ZFS. Cette configuration n'est pas recommandée pour les systèmes de production, mais peut être utilisée pour les besoins en transition ou migration sur un petit système, tel qu'un ordinateur portable.

Problèmes JumpStart pour 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
```

- Vous devez créer le pool avec des tranches de disque plutôt qu'avec des disques entiers, comme décrit à la section [“Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS”](#) à la page 127. Par exemple, la syntaxe en gras dans l'exemple suivant n'est pas correcte :

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

La syntaxe en gras dans l'exemple suivant est correcte :

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

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

Les fonctions d'Oracle Solaris Live Upgrade concernant les composants UFS sont toujours disponibles et fonctionnent comme dans les versions précédentes de Solaris.

Les fonctions suivantes sont également 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 faire migrer un système comportant des zones, mais les configurations prises en charge sont limitées dans la version 10 10/08 de Solaris. À compter de la version 10 5/09, Solaris prend en charge davantage de configurations de zone. Pour plus d'informations, consultez les sections suivantes :
 - [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones \(Solaris 10 10/08\)”](#) à la page 151
 - [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones \(version Solaris 5 10/09 ou supérieure\)”](#) à la page 156

Si vous migrez vers un système sans zones, reportez-vous à la section [“Utilisation d'Oracle Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS \(sans zones\)”](#) à la page 146.

- Lorsque vous créez un environnement d'initialisation ZFS dans le même pool, Oracle Solaris Live Upgrade peut faire appel aux fonctions d'instantané et de clonage ZFS. La création d'un environnement d'initialisation est de ce fait plus rapide que dans les versions précédentes de Solaris.

Pour plus d'informations sur l'installation d'Oracle Solaris et sur les fonctions d'Oracle Solaris Live Upgrade, reportez-vous au [Guide d'installation Oracle Solaris 10 9/10 : planification des mises à niveau et de Solaris Live Upgrade](#).

Le processus de base de la migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS est le suivant :

- Installez Solaris 10 10/08, 10 5/09, 10 10/09, 10 9/10 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 la version Solaris 10 10/08 ou une version supérieure, créez un pool de stockage ZFS pour votre système de fichiers racine ZFS.
- Utilisez le composant Oracle Solaris Live Upgrade pour migrer votre système de fichiers UFS racine vers un système de fichiers ZFS racine.
- Activez votre environnement d'initialisation ZFS à l'aide de la commande `luactivate`.

Pour plus d'informations sur la configuration requise de ZFS et d'Oracle Solaris Live Upgrade, reportez-vous à la section [“Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS”](#) à la page 127.

Problèmes de migration ZFS avec Oracle Solaris Live Upgrade

Vérifiez la liste de problèmes suivants avant d'utiliser Oracle 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 du programme d'installation d'Oracle 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 Oracle Solaris Live Upgrade.
- Avant de démarrer Oracle Solaris Live Upgrade, vous devez créer le pool de stockage ZFS à utiliser pour l'initialisation. 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, assurez-vous que le processus d'étiquetage n'a pas modifié le schéma de partitionnement. Dans la plupart des cas, toute la capacité du disque doit se trouver dans les tranches destinées au pool racine.

- Vous ne pouvez pas utiliser Oracle Solaris Live Upgrade pour créer un environnement d'initialisation UFS à partir d'un environnement d'initialisation ZFS. Si vous migrez votre environnement d'initialisation UFS vers un environnement d'initialisation ZFS tout en conservant votre environnement d'initialisation UFS, vous pouvez initialiser l'un ou l'autre des environnements (UFS ou ZFS).
- Ne renommez pas vos environnements d'initialisation ZFS à l'aide de la commande `zfs rename` car la fonction Oracle Solaris Live Upgrade ne détecte pas les changements de nom. Toute commande utilisée ultérieurement, notamment `ludelete`, échoue. Ne renommez en fait pas vos pools ZFS ni vos systèmes de fichiers ZFS si vous disposez d'environnements d'initialisation que vous souhaitez continuer à utiliser.
- Lorsque vous créez un 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 Oracle 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 d'Oracle 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 d'Oracle 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 comportant des zones ou effectuez sa mise à jour, reportez-vous aux sections suivantes :

- [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones \(Solaris 10 10/08\)”](#) à la page 151
- [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones \(version Solaris 5 10/09 ou supérieure\)”](#) à la page 156

EXEMPLE 5-3 Utilisation d'Oracle Solaris Live Upgrade pour migrer d'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, `ufsBE`, 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, `zfsBE`, est identifié par l'option `-n`. L'utilisation de la commande `lucreate` requiert l'existence préalable d'un pool de stockage ZFS.

Pour pouvoir mettre à niveau et initialiser le pool de stockage ZFS, vous devez le créer avec des tranches plutôt que des disques entiers. Avant de créer le nouveau pool, assurez-vous que les disques à utiliser dans le pool portent une étiquette SMI (VTOC) au lieu d'une étiquette EFI. Si le disque a été réétiqueté avec une étiquette SMI, assurez-vous que le processus d'étiquetage n'a pas modifié le schéma de partitionnement. Dans la plupart des cas, toute la capacité du disque doit se trouver dans la tranche destinée au pool racine.

```
# zpool create rpool mirror clt2d0s0 c2t1d0s0
# lucreate -c ufsBE -n zfsBE -p rpool
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <ufsBE>.
Creating initial configuration for primary boot environment <ufsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/clt2d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Populating file systems on boot environment <zfsBE>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Creating compare databases for boot environment <zfsBE>.
Creating compare database for file system </rpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfsBE>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-qD.mnt
updating /.alt.tmp.b-qD.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
```

EXEMPLE 5-3 Utilisation d'Oracle Solaris Live Upgrade pour migrer d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Suite)

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
-----
ufsBE                  yes     yes   yes     no    -
zfsBE                  yes     no    no      yes   -
```

Examinez ensuite la liste des composants ZFS. Exemple :

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                7.17G 59.8G 95.5K  /rpool
rpool/ROOT           4.66G 59.8G 21K   /rpool/ROOT
rpool/ROOT/zfsBE    4.66G 59.8G 4.66G /
rpool/dump           2G    61.8G 16K   -
rpool/swap           517M  60.3G 16K   -
```

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

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

```
*****
```

```
The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.
```

```
*****
```

```
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
```

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

```
# init 6
```

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

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
```

EXEMPLE 5-3 Utilisation d'Oracle Solaris Live Upgrade pour migrer d'un système de fichiers racine UFS vers un système de fichiers racine ZFS (Suite)

```
-----
ufsBE          yes    no    no    yes    -
zfsBE          yes    yes   yes   no     -
```

Si vous repassez à l'environnement d'initialisation UFS, vous devez réimporter tous les pools de stockage ZFS créés au cours de l'initialisation de l'environnement d'initialisation ZFS car ils ne sont pas disponibles automatiquement dans l'environnement d'initialisation UFS.

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

EXEMPLE 5-4 Utilisation d'Oracle 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 actuel se trouve sur le même pool ZFS, l'option `-p` est omise.

Si vous disposez de plusieurs environnements d'initialisation ZFS, procédez comme suit pour sélectionner l'environnement d'initialisation à partir duquel vous voulez initialiser le système :

- SPARC : vous pouvez utiliser la commande `boot -L` pour identifier les environnements d'initialisation disponibles et en sélectionner un à l'aide de la commande `boot -Z`.
- x86 : vous pouvez sélectionner un environnement d'initialisation à partir du menu GRUB.

Pour plus d'informations, reportez-vous à l'[Exemple 5-9](#).

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
```

EXEMPLE 5-4 Utilisation d'Oracle Solaris Live Upgrade pour créer un environnement d'initialisation ZFS à partir d'un environnement d'initialisation ZFS (Suite)

Creation of boot environment <zfs2BE> successful.

EXEMPLE 5-5 Mise à niveau de votre environnement d'initialisation ZFS (luupgrade)

Vous pouvez mettre à niveau votre environnement d'initialisation ZFS à l'aide de packages ou de patchs supplémentaires.

Le processus de base est le suivant :

- Créez un environnement d'initialisation alternatif à l'aide de la commande `lucreate`.
- Activez et initialisez le système à partir de l'environnement d'initialisation alternatif.
- Mettez votre environnement d'initialisation ZFS principal à niveau à l'aide de la commande `luupgrade` pour ajouter des packages ou des patchs.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                 Complete Now    On Reboot Delete Status
-----
zfsBE                 yes     no     no      yes   -
zfs2BE                yes     yes    yes     no    -
# luupgrade -p -n zfsBE -s /net/system/export/s10up/Solaris_10/Product SUNWchxge
Validating the contents of the media </net/install/export/s10up/Solaris_10/Product>.
Mounting the BE <zfsBE>.
Adding packages to the BE <zfsBE>.

Processing package instance <SUNWchxge> from </net/install/export/s10up/Solaris_10/Product>

Chelsio N110 10GE NIC Driver(sparc) 11.10.0,REV=2006.02.15.20.41
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.

This appears to be an attempt to install the same architecture and
version of a package which is already installed. This installation
will attempt to overwrite this package.

Using </a> as the package base directory.
## Processing package information.
## Processing system information.
   4 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SUNWchxge> [y,n,?] y
Installing Chelsio N110 10GE NIC Driver as <SUNWchxge>

## Installing part 1 of 1.
## Executing postinstall script.
```

EXEMPLE 5-5 Mise à niveau de votre environnement d'initialisation ZFS (Luupgrade) (Suite)

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

Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones (Solaris 10 10/08)

Vous pouvez utiliser Oracle Solaris Live Upgrade pour migrer un système avec zones. Dans Solaris 10 10/08, les configurations prises en charge sont toutefois limitées. Si vous installez Solaris 10 5/09 ou effectuez une mise à niveau vers cette version, un nombre plus important de configurations de zone est pris en charge. Pour plus d'informations, reportez-vous à la section “Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 5 10/09 ou supérieure)” à la page 156.

Cette section explique comment configurer et installer un système avec zones de manière à ce qu'il soit mis à jour et patché avec Oracle Solaris Live Upgrade. Si vous migrez vers un système de fichiers racine ZFS sans zones, reportez-vous à la section “[Utilisation d'Oracle Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS \(sans zones\)](#)” à la page 146.

Si vous migrez un système avec zones ou configurez un système avec zones dans Solaris 10 10/08, consultez les procédures suivantes :

- “[Migration d'un système de fichiers racine UFS avec racines de zone sur UFS vers un système de fichiers racine ZFS \(Solaris 10 10/08\)](#)” à la page 152
- “[Configuration d'un système de fichiers racine ZFS avec racines de zone sur ZFS \(Solaris 10 10/08\)](#)” à la page 154
- “[Mise à niveau ou application de patch à un système de fichiers racine ZFS avec racines de zone sur ZFS \(Solaris 10 10/08\)](#)” à la page 155
- “[Résolution de problèmes de point de montage empêchant l'initialisation \(Solaris 10 10/08\)](#)” à la page 175

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

▼ Migration d'un système de fichiers racine UFS avec racines de zone sur UFS vers un système de fichiers racine ZFS (Solaris 10 10/08)

La procédure suivante explique comment migrer d'un système de fichiers racine UFS comportant des zones installées vers un système de fichiers racine ZFS et une configuration de racine de zone ZFS pouvant être mis à niveau ou patchés.

Dans les étapes suivantes de cet exemple, le pool porte le nom `rpool` et l'environnement d'initialisation actif porte le nom `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 Oracle Solaris 10 9/10 : planification des mises à niveau et de Solaris Live Upgrade*.

2 Créez le pool racine.

```
# zpool create rpool mirror c0t1d0 c1t1d0
```

Pour plus d'informations sur la configuration requise pour le pool racine, reportez-vous à la section “Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS” à la page 127.

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

4 Créez le nouvel environnement d'initialisation ZFS.

```
# lucreate -n s10BE2 -p rpool
```

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

```
# luactivate s10BE2
```

Le système exécute désormais 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 ZFS 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 Résolvez les éventuels problèmes de point de montage.

Étant donné la présence d'un bogue dans Oracle Solaris Live Upgrade, il se peut que l'environnement d'initialisation inactif ne puisse pas s'initialiser. Ce problème est lié à la présence d'un point de montage non valide dans un jeu de données ZFS ou dans un jeu de données ZFS d'une zone de l'environnement d'initialisation.

a. Contrôlez la sortie `zfs list`.

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

▼ Configuration d'un système de fichiers racine ZFS avec racines de zone sur ZFS (Solaris 10 10/08)

La procédure suivante explique comment installer un système de fichiers racine ZFS et une configuration racine de zone ZFS pouvant être mis à niveau ou patchés. Dans cette configuration, les racines de zone ZFS sont créées sous forme de jeux de données ZFS.

Dans les étapes suivantes de cet exemple, le pool porte le nom `rpool` et l'environnement d'initialisation actif porte le nom `S10be`. Le nom du jeu de données de zones peut être tout nom de jeu de données légal. Dans l'exemple suivant, le jeu de données des zones porte le nom `zones`.

- 1 **Installez le système avec une racine ZFS en utilisant soit le programme d'installation en mode texte interactif Solaris, 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 Solaris JumpStart, reportez-vous à la section [“Installation d'un système de fichiers racine ZFS \(installation initiale\)”](#) à la page 130 ou [“Installation d'un système de fichiers racine ZFS \(installation d'Oracle Solaris JumpStart\)”](#) à la page 140.

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

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

Exemple :

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones
```

La définition de la valeur `noauto` de la propriété `canmount` permet d'éviter que le jeu de données ne soit monté d'une autre manière que par l'action explicite d'Oracle Solaris Live Upgrade et le code de démarrage du système.

- 4 **Montez le jeu de données de zones créé.**

```
# zfs mount rpool/ROOT/s10BE/zones
```

Le jeu de données est monté sous `/zones`.

- 5 **Créez et montez un jeu de données pour chaque 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
```

▼ Mise à niveau ou application de patch à un système de fichiers racine ZFS avec racines de zone sur ZFS (Solaris 10 10/08)

Suivez cette procédure pour mettre à niveau ou patcher un système de fichiers ZFS racine comportant des racines de zone. Ces mises à jour peuvent être une mise à niveau du système ou l'application de correctifs.

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

1 Créez l'environnement d'initialisation à mettre à jour ou à corriger.

```
# lucreate -n newBE
```

L'environnement d'initialisation existant, y compris toutes les zones, est cloné. Chaque jeu de données de l'environnement d'initialisation d'origine est créé. 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.

```
# luactivate newBE
```

4 Initialisez le système à partir de l'environnement d'initialisation récemment activé.

```
# init 6
```

5 Réolvez les éventuels problèmes de point de montage.

Étant donné la présence d'un bogue dans Oracle Solaris Live Upgrade, il se peut que l'environnement d'initialisation inactif ne puisse pas s'initialiser. Ce problème est lié à la présence d'un point de montage non valide dans un jeu de données ZFS ou dans un jeu de données ZFS d'une zone de l'environnement d'initialisation.

a. Contrôlez la sortie `zfs list`.

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

Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 5 10/09 ou supérieure)

À partir de la version 10 10/08 de Solaris, vous pouvez utiliser la fonctionnalité Oracle Solaris Live Upgrade pour migrer ou mettre à niveau un système avec zones. À partir de la version 10 5/09 de Solaris, Oracle Solaris Live Upgrade prend en charge des configurations de zone complète ou fragmentée.

Cette section décrit comment configurer et installer un système avec zones de manière à ce qu'il soit mis à niveau et corrigé avec Oracle Solaris Live Upgrade avec la version 10 5/09. Si vous

migrez vers un système de fichiers racine ZFS sans zones, reportez-vous à la section [“Utilisation d'Oracle Solaris Live Upgrade pour migrer vers un système de fichiers racine ZFS \(sans zones\)”](#) à la page 146.

Tenez compte des points suivants lorsque vous utilisez Oracle Solaris Live Upgrade avec ZFS et les zones à partir de la version Solaris 10 5/09 :

- À partir de la version 10 5/09 de Solaris, pour utiliser Oracle Solaris Live Upgrade avec des configurations de zone prises en charge, vous devez au préalable mettre à niveau votre système vers la version 10 5/09 de Solaris ou une version supérieure à l'aide du programme de mise à niveau standard.
- Puis, avec Oracle Solaris Live Upgrade, vous pouvez soit migrer votre système de fichiers racine UFS avec racines de zone vers un système de fichiers racine ZFS, soit mettre à niveau votre système de fichiers racine ZFS et vos racines de zone ou leur appliquer un patch.
- Vous ne pouvez pas migrer les configurations de zones non prises en charge à partir d'une version antérieure à Solaris 10 vers la version 10 5/09 de Solaris ou une version supérieure.

À partir de la version 10 5/09 de Solaris, si vous migrez ou configurez un système comportant des zones, vous devez vérifier les informations suivantes :

- [“Système de fichiers ZFS pris en charge avec informations de configuration de la racine de zone \(version Solaris 10 5/09 ou supérieure\)”](#) à la page 157
- [“Création d'un environnement d'initialisation ZFS avec un système de fichiers racine ZFS et une zone racine \(Solaris 10 5/09 ou version ultérieure\)”](#) à la page 159
- [“Mise à niveau ou correction d'un système de fichiers racine ZFS avec racines de zone \(Solaris 10 5/09 ou version ultérieure\)”](#) à la page 161
- [“Migration d'un système de fichiers racine UFS avec racine de zone vers un système de fichiers racine ZFS \(Solaris 10 5/09 ou version ultérieure\)”](#) à la page 164

Système de fichiers ZFS pris en charge avec informations de configuration de la racine de zone (version Solaris 10 5/09 ou supérieure)

Vérifiez les configurations de zone prises en charge avant d'utiliser Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones.

- **Migration d'un système de fichiers racine UFS vers un système de fichiers racine ZFS.** Les configurations de racine de zone suivantes sont prises en charge :
 - Dans un répertoire du système de fichiers racine UFS
 - Dans le sous-répertoire d'un point de montage, dans le système de fichiers racine UFS
 - Système de fichiers racine UFS avec une racine de zone se trouvant dans le répertoire d'un système de fichiers racine UFS ou dans le sous-répertoire d'un point de montage d'un système de fichiers racine UFS et un pool ZFS non racine avec une racine de zone.

La configurations UFS/zone suivante n'est pas prise en charge : système de fichiers racine UFS avec racine de zone en tant que point de montage.

- **Migration ou mise à niveau d'un système de fichiers racine ZFS.** Les configurations de racine de zone suivantes sont prises en charge :
 - Dans un jeu de données dans le pool racine ZFS. Dans certains cas, si un jeu de données n'est pas fourni pour la racine de zone avant l'opération Oracle Solaris Live Upgrade, un jeu de données pour la racine de zone (zoneds) est créé par Oracle Solaris Live Upgrade.
 - Dans un sous-répertoire du système de fichiers racine ZFS
 - Dans un jeu de données ne faisant pas partie du système de fichiers racine ZFS
 - Dans le sous-répertoire d'un jeu de données ne faisant pas partie du système de fichiers racine ZFS
 - Dans un jeu de données d'un pool non-racine. Dans l'exemple suivant, zonepool/zones est un jeu de données qui contient les racines de zone et rpool contient l'environnement d'initialisation ZFS :

```
zonepool
zonepool/zones
zonepool/zones/myzone
rpool
rpool/ROOT
rpool/ROOT/myBE
```

Oracle Solaris Live Upgrade prend un instantané et clone les zones contenues dans zonepool et dans l'environnement d'initialisation rpool, si vous utilisez la syntaxe suivante :

```
# lucreate -n newBE
```

L'environnement d'initialisation newBE sous rpool/root/newBE est créé. Lorsque ce dernier est activé, l'environnement d'initialisation newBE fournit l'accès aux composants zonepool.

Dans l'exemple, si /zonepool/zones était un sous-répertoire et non un autre jeu de données, Live Upgrade le migrerait comme composant du pool racine, rpool.

- **Migration de zones ou informations de mise à niveau avec zones pour les environnements UFS et ZFS :** vérifiez les éléments suivants, car ils sont susceptibles d'affecter les migrations ou les mises à jour d'un environnement ZFS ou UFS :
 - Si vous avez configuré les zones comme décrit à la section [“Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones \(Solaris 10 10/08\)”](#) à la page 151 dans la version Solaris 10 10/08 et que vous avez effectué une mise à niveau au moins vers la version Solaris 10 5/09, vous devriez pouvoir opérer une migration vers un système de fichiers racine ZFS ou bien une mise à niveau vers Solaris 10 5/09 à l'aide de Solaris Live Upgrade.

- Ne créez aucune racine de zone dans les répertoires imbriqués, par exemple : zones/zone1 et zones/zone1/zone2. Sinon, le montage risque d'échouer lors de l'initialisation.

▼ Création d'un environnement d'initialisation ZFS avec un système de fichiers racine ZFS et une zone racine (Solaris 10 5/09 ou version ultérieure)

Après avoir procédé à l'installation initiale de la version Solaris 10 5/09 ou d'une version supérieure, suivez cette procédure pour créer un système de fichiers ZFS racine. Suivez également la procédure suivante si vous avez utilisé la fonction `luupgrade` pour mettre à niveau un système de fichiers racine ZFS vers la version 10 5/09 ou une version supérieure de Solaris. Un environnement d'initialisation ZFS créé à l'aide de cette procédure peut être mis à niveau ou corrigé à l'aide d'un patch.

Dans la procédure ci-après, l'exemple de système Oracle Solaris 10 9/10 comporte un système de fichiers racine ZFS et un jeu de données de racine de zone dans `/rpool/zones`. Un environnement d'initialisation ZFS nommé `zfs2BE` est créé et peut ensuite être mis à niveau ou corrigé à l'aide d'un patch.

1 Vérifiez les systèmes de fichiers ZFS existants.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                7.26G 59.7G   98K    /rpool
rpool/ROOT                           4.64G 59.7G   21K    legacy
rpool/ROOT/zfsBE                     4.64G 59.7G  4.64G    /
rpool/dump                           1.00G 59.7G  1.00G    -
rpool/export                         44K   59.7G   23K    /export
rpool/export/home                    21K   59.7G   21K    /export/home
rpool/swap                           1G    60.7G  16K    -
rpool/zones                          633M  59.7G  633M    /rpool/zones
```

2 Assurez-vous que les zones sont installées et initialisées.

```
# zoneadm list -cv
ID NAME          STATUS  PATH                BRAND  IP
0  global         running /                   native shared
2  zfszone        running /rpool/zones       native shared
```

3 Créez l'environnement d'initialisation ZFS.

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
```

```

system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.

```

4 Activez l'environnement d'initialisation ZFS.

```

# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                 Complete Now    On Reboot Delete Status
-----
zfsBE                 yes     yes   yes     no    -
zfs2BE                yes     no    no      yes   -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
# init 6

```

5 Confirmez que les systèmes de fichiers ZFS et les zones sont créés dans le nouvel environnement d'initialisation.

```

# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               7.38G 59.6G  98K    /rpool
rpool/ROOT                          4.72G 59.6G   21K    legacy
rpool/ROOT/zfs2BE                   4.72G 59.6G  4.64G   /
rpool/ROOT/zfs2BE@zfs2BE            74.0M  -      4.64G   -
rpool/ROOT/zfsBE                    5.45M 59.6G  4.64G   /.alt.zfsBE
rpool/dump                          1.00G 59.6G  1.00G   -
rpool/export                       44K   59.6G  23K    /export
rpool/export/home                   21K   59.6G  21K    /export/home
rpool/swap                          1G    60.6G  16K    -
rpool/zones                         17.2M 59.6G  633M   /rpool/zones
rpool/zones-zfsBE                   653M  59.6G  633M   /rpool/zones-zfsBE
rpool/zones-zfsBE@zfs2BE           19.9M  -      633M   -
# zoneadm list -cv
ID  NAME          STATUS  PATH          BRAND  IP
0   global       running /              native shared
-   zfszone     installed /rpool/zones  native shared

```


▼ Mise à niveau ou correction d'un système de fichiers racine ZFS avec racines de zone (Solaris 10 5/09 ou version ultérieure)

Suivez cette procédure pour mettre à niveau ou corriger un système de fichiers ZFS racine avec racines de zone sur Solaris 10 5/09 ou version supérieure. Ces mises à jour peuvent être une mise à niveau du système ou l'application de correctifs.

Dans les étapes suivantes, `zfs2BE` est le nom de l'environnement d'initialisation mis à niveau ou corrigé.

1 Vérifiez les systèmes de fichiers ZFS existants.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                7.38G 59.6G  100K   /rpool
rpool/ROOT                            4.72G 59.6G   21K   legacy
rpool/ROOT/zfs2BE                     4.72G 59.6G  4.64G   /
rpool/ROOT/zfs2BE@zfs2BE              75.0M  -      4.64G   -
rpool/ROOT/zfsBE                      5.46M 59.6G  4.64G   /
rpool/dump                            1.00G 59.6G  1.00G   -
rpool/export                          44K   59.6G  23K    /export
rpool/export/home                     21K   59.6G  21K    /export/home
rpool/swap                             1G    60.6G  16K    -
rpool/zones                           22.9M 59.6G  637M   /rpool/zones
rpool/zones-zfsBE                     653M 59.6G  633M   /rpool/zones-zfsBE
rpool/zones-zfsBE@zfs2BE              20.0M  -      633M   -
```

2 Assurez-vous que les zones sont installées et initialisées.

```
# zoneadm list -cv
ID NAME          STATUS  PATH                               BRAND  IP
  0 global        running /                                   native shared
  5 zfszone        running /rpool/zones                       native shared
```

3 Créez l'environnement d'initialisation ZFS à mettre à jour ou à corriger.

```
# lucreate -n zfs2BE
Analyzing system configuration.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Creating snapshot for <rpool/zones> on <rpool/zones@zfs10092BE>.
Creating clone for <rpool/zones@zfs2BE> on <rpool/zones-zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

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

- Mettez à niveau le système.

```
# luupgrade -u -n zfs2BE -s /net/install/export/s10up/latest
```

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

Ce processus peut être très long.

Pour un exemple complet du processus luupgrade, reportez-vous à l'[Exemple 5-6](#).

- Appliquez les correctifs au nouvel environnement d'initialisation.

```
# luupgrade -t -n zfs2BE -t -s /patchdir patch-id-02 patch-id-04
```

5 Activez le nouvel environnement d'initialisation.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
zfsBE                  yes     yes   yes     no    -
zfs2BE                 yes     no    no      yes   -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
```

6 Initialisez le système à partir de l'environnement d'initialisation récemment activé.

```
# init 6
```

Exemple 5-6 Mise à niveau d'un système de fichiers racine ZFS avec racine de zone vers un système de fichiers racine ZFS Oracle Solaris 10 9/10

Dans cet exemple, un environnement d'initialisation ZFS (zfsBE), créé sur un système Solaris 10 10/09 avec un système de fichiers racine ZFS et une racine de zone dans un pool non-racine, est mis à niveau vers Solaris 10 9/10. Ce processus peut être long. L'environnement d'initialisation mis à niveau (zfs2BE) est ensuite activé. Assurez-vous que les zones sont installées et initialisées avant de tenter la mise à niveau.

Dans cet exemple, le pool zonepool, le jeu de données /zonepool/zones et la zone zfszone sont créés comme suit :

```
# zpool create zonepool mirror c2t1d0 c2t5d0
# zfs create zonepool/zones
# chmod 700 zonepool/zones
# zonecfg -z zfszone
zfszone: No such zone configured
Use 'create' to begin configuring a new zone.
```

```

zonecfg:zfszone> create
zonecfg:zfszone> set zonepath=/zonepool/zones
zonecfg:zfszone> verify
zonecfg:zfszone> exit
# zoneadm -z zfszone install
cannot create ZFS dataset zonepool/zones: dataset already exists
Preparing to install zone <zfszone>.
Creating list of files to copy from the global zone.
Copying <8960> files to the zone.
.
.
.

# zoneadm list -cv
ID NAME           STATUS   PATH                      BRAND  IP
  0 global          running  /                          native shared
  2 zfszone        running  /zonepool/zones           native shared

# lucreate -n zfsBE
.
.
.
# luupgrade -u -n zfsBE -s /net/install/export/s10up/latest
40410 blocks
miniroot filesystem is <lofs>
Mounting miniroot at </net/system/export/s10up/latest/Solaris_10/Tools/Boot>
Validating the contents of the media </net/system/export/s10up/latest>.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains <Solaris> version <10>.
Constructing upgrade profile to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE <zfsBE>.
Determining packages to install or upgrade for BE <zfsBE>.
Performing the operating system upgrade of the BE <zfsBE>.
CAUTION: Interrupting this process may leave the boot environment unstable
or unbootable.
Upgrading Solaris: 100% completed
Installation of the packages from this media is complete.
Updating package information on boot environment <zfsBE>.
Package information successfully updated on boot environment <zfsBE>.
Adding operating system patches to the BE <zfsBE>.
The operating system patch installation is complete.
INFORMATION: The file </var/sadm/system/logs/upgrade_log> on boot
environment <zfsBE> contains a log of the upgrade operation.
INFORMATION: The file </var/sadm/system/data/upgrade_cleanup> on boot
environment <zfsBE> contains a log of cleanup operations required.
INFORMATION: Review the files listed above. Remember that all of the files
are located on boot environment <zfsBE>. Before you activate boot
environment <zfsBE>, determine if any additional system maintenance is
required or if additional media of the software distribution must be
installed.
The Solaris upgrade of the boot environment <zfsBE> is complete.
Installing failsafe
Failsafe install is complete.
# luactivate zfsBE
# init 6

```

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
zfsBE                 yes     no     no      yes   -
zfs2BE                yes     yes    yes     no    -
# zoneadm list -cv
ID NAME                STATUS  PATH                                BRAND  IP
0  global               running /                                    native shared
-  zfszone              installed /zonepool/zones                  native shared
```

▼ Migration d'un système de fichiers racine UFS avec racine de zone vers un système de fichiers racine ZFS (Solaris 10 5/09 ou version ultérieure)

Suivez cette procédure pour migrer un système incluant un système de fichiers UFS racine et une racine de zone vers la version Solaris 10 5/09 ou version supérieure. Utilisez ensuite Oracle Solaris Live Upgrade pour créer un environnement d'initialisation ZFS.

Dans la procédure ci-après, l'exemple de nom de l'environnement d'initialisation UFS est `c0t1d0s0`, la racine de zone UFS est `zonepool/zfszone` et l'environnement d'initialisation racine est `zfsBE`.

1 Mettez le système à niveau vers la version Solaris 10 5/09 ou un version ultérieure si la version Solaris 10 exécutée est antérieure.

Pour plus d'informations sur la mise à niveau d'un système exécutant Solaris 10, reportez-vous au *Guide d'installation Oracle Solaris 10 9/10 : planification des mises à niveau et de Solaris Live Upgrade*.

2 Créez le pool racine.

Pour plus d'informations sur la configuration requise pour le pool racine, reportez-vous à la section “[Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS](#)” à la page 127.

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

```
# zoneadm list -cv
ID NAME                STATUS  PATH                                BRAND  IP
0  global               running /                                    native shared
2  zfszone              running /zonepool/zones                  native shared
```

4 Créez le nouvel environnement d'initialisation ZFS.

```
# lucreate -c c1t1d0s0 -n zfsBE -p rpool
```

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

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
c1t1d0s0              yes     no     no      yes   -
zfsBE                  yes     yes    yes     no    -      #
Luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.
.
.
.
```

6 Redémarrez le système.

```
# init 6
```

7 Confirmez que les systèmes de fichiers ZFS et les zones sont créés dans le nouvel environnement d'initialisation.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                6.17G 60.8G   98K    /rpool
rpool/ROOT                           4.67G 60.8G   21K    /rpool/ROOT
rpool/ROOT/zfsBE                      4.67G 60.8G   4.67G  /
rpool/dump                            1.00G 60.8G   1.00G  -
rpool/swap                            517M 61.3G   16K    -
zonepool                              634M 7.62G   24K    /zonepool
zonepool/zones                       270K 7.62G   633M   /zonepool/zones
zonepool/zones-c1t1d0s0              634M 7.62G   633M   /zonepool/zones-c1t1d0s0
zonepool/zones-c1t1d0s0@zfsBE       262K  -      633M   -
# zoneadm list -cv
ID NAME          STATUS  PATH                                BRAND  IP
0 global        running /                                     native shared
- zfszone      installed /zonepool/zones                    native  shared
```

Exemple 5-7 Migration d'un système de fichiers racine UFS avec racine de zone vers un système de fichiers racine ZFS

Dans cet exemple, un système Oracle Solaris 10 9/10 comprenant un système de fichiers racine UFS, une racine de zone (/uzone/ufszone), un pool non-racine ZFS (pool) et une racine de zone (/pool/zfszone) est migré vers un système de fichiers racine ZFS. Assurez-vous que le pool racine ZFS est créé et que les zones sont installées et initialisées avant de tenter la migration.

```
# zoneadm list -cv
ID NAME          STATUS  PATH                                BRAND  IP
0 global        running /                                     native shared
2 ufszone      running /uzone/ufszone                    native shared
3 zfszone      running /pool/zones/zfszone                native  shared
```

```
# lucreate -c ufsBE -n zfsBE -p rpool
Analyzing system configuration.
```

```

No name for current boot environment.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Populating file systems on boot environment <zfsBE>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Copying root of zone <ufszone> to </.alt.tmp.b-EYd.mnt/uzone/ufszone>.
Creating snapshot for <pool/zones/zfszone> on <pool/zones/zfszone@zfsBE>.
Creating clone for <pool/zones/zfszone@zfsBE> on <pool/zones/zfszone-zfsBE>.
Creating compare databases for boot environment <zfsBE>.
Creating compare database for file system </rpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfsBE>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-DLd.mnt
updating /.alt.tmp.b-DLd.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
# lustatus
Boot Environment      Is      Active Active   Can      Copy
Name                  Complete Now   On Reboot Delete Status
-----
ufsBE                  yes     yes   yes     no       -
zfsBE                  yes     no    no      yes      -
# luactivate zfsBE
.
.
.
# init 6
.
.
.
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
pool                 628M  66.3G  19K    /pool
pool/zones           628M  66.3G  20K    /pool/zones
pool/zones/zfszone   75.5K  66.3G  627M   /pool/zones/zfszone
pool/zones/zfszone-ufsBE  628M  66.3G  627M   /pool/zones/zfszone-ufsBE
pool/zones/zfszone-ufsBE@zfsBE  98K   -      627M   -
rpool                7.76G  59.2G  95K    /rpool
rpool/ROOT           5.25G  59.2G  18K    /rpool/ROOT
rpool/ROOT/zfsBE    5.25G  59.2G  5.25G  /

```

```

rpool/dump                2.00G  59.2G  2.00G  -
rpool/swap                517M  59.7G   16K  -
# zoneadm list -cv
ID NAME          STATUS   PATH                                BRAND  IP
0 global        running  /                                    native shared
- ufszone       installed /uzone/ufszone                     native shared
- zfszone       installed /pool/zones/zfszone                native shared

```

Prise en charge ZFS des périphériques de swap et de vidage

Lors de l'installation initiale d'un SE Solaris ou après la migration d'Oracle 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. Par exemple :

```

# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 256,1    16 4194288 4194288

```

Lors de l'installation initiale d'un SE Solaris ou de l'utilisation d'Oracle 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. En règle générale, un périphérique de vidage ne nécessite aucune administration car il est créé automatiquement lors de l'installation. Exemple :

```

# dumpadm
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
Savecore enabled: yes
Save compressed: on

```

Si vous désactivez et supprimez le périphérique de vidage, vous devrez l'activer avec la commande `dumpadm` après sa recréation. Dans la plupart des cas, vous devrez uniquement ajuster la taille du périphérique de vidage à l'aide de la commande `zfs`.

Pour plus d'informations sur les tailles des volumes de swap et de vidage créés par les programmes d'installation, reportez-vous à la section [“Configuration requise pour l'installation d'Oracle Solaris et d'Oracle Solaris Live Upgrade pour la prise en charge de systèmes de fichiers ZFS”](#) à la page 127.

La taille des volume d'échange et de vidage peut être ajustée pendant et après l'installation. Pour plus d'informations, reportez-vous à la section [“Ajustement de la taille de vos périphériques de swap et de vidage ZFS”](#) à la page 168.

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 la zone de swap et le périphérique de vidage.

- L'utilisation d'un fichier swap sur un système de fichiers ZFS n'est actuellement pas prise en charge.
- Pour modifier la zone de swap ou le périphérique de vidage une fois le système installé ou mis à niveau, utilisez les commandes `swap` et `dumpadm` de la même façon que dans les versions 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).

Pour de plus amples informations, reportez-vous aux sections suivantes :

- “Ajustement de la taille de vos périphériques de swap et de vidage ZFS” à la page 168
- “Dépannage du périphérique de vidage ZFS” à la page 170

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

La façon dont une installation racine ZFS détermine la taille des périphériques de swap et de vidage diffère ; 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 5-1](#).
- Vous pouvez créer des volumes de swap et de vidage, ainsi que leur attribuer une taille, avant de procéder à une opération Oracle Solaris Live Upgrade. Exemple :

1. Créer le pool de stockage.

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

2. Créez le périphérique de vidage.

```
# zfs create -V 2G rpool/dump
```

3. Activez le périphérique de vidage.

```
# dumpadm -d /dev/zvol/dsk/rpool/dump
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
Savecore enabled: yes
Save compressed: on
```

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

- SPARC : créez votre zone de swap. Configurez la taille du bloc sur 8 Ko.

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

- x86 : créez votre zone de swap. Configurez la taille du bloc sur 4 Ko.


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

5. Vous devez activer le volume de swap lorsqu'un nouveau périphérique de swap est ajouté ou modifié.
6. Ajoutez une entrée pour le volume d'échange dans le fichier `/etc/vfstab`.

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

- Vous pouvez rétablir la propriété `volsize` du périphérique de vidage après l'installation d'un système. Exemple :

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

- Vous pouvez redimensionner le volume de swap, mais tant que le problème CR 6765386 existe, il est préférable de supprimer au préalable le périphérique de swap. Recréez-le par la suite. Exemple :

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs volsize=2G rpool/swap
# swap -a /dev/zvol/dsk/rpool/swap
```

Pour plus d'informations sur la suppression d'un périphérique de swap sur un système actif, reportez-vous au site suivant :

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

- Vous pouvez ajuster la taille des volumes de swap et de vidage d'un profil JumpStart à l'aide d'une syntaxe de profil semblable à la suivante :

```
install_type initial_install
cluster SUNWCxall
pool rpool 16g 2g 2g c0t0d0s0
```

Dans ce profil, les deux entrées `2g` définissent respectivement la taille du volume de swap et de vidage sur 2 Go.

- Si vous avez besoin de plus d'espace de swap sur un système déjà installé, il suffit d'ajouter un autre volume de swap. Exemple :

```
# zfs create -V 2G rpool/swap2
```

Activez ensuite le nouveau volume de swap. Exemple :

```
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap  256,1    16 1058800 1058800
/dev/zvol/dsk/rpool/swap2 256,3    16 4194288 4194288
```

Ajoutez ensuite une entrée pour le deuxième volume de swap dans le fichier `/etc/vfstab`.

Dépannage du périphérique de vidage ZFS

Vérifiez les éléments suivants si vous rencontrez des problèmes soit lors de la capture d'un vidage mémoire sur incident du système, soit lors du redimensionnement du périphérique de vidage.

- Si un vidage mémoire sur incident n'a pas été automatiquement créé, vous pouvez utiliser la commande `savecore` pour enregistrer le vidage mémoire sur incident.
- Lorsque vous installez un système de fichiers ZFS racine ou lorsque vous effectuez une migration vers un système de fichiers ZFS racine pour la première fois, un volume de vidage est automatiquement créé. Dans la plupart des cas, vous devrez uniquement ajuster la taille par défaut du périphérique de vidage si celle-ci est trop petite. Par exemple, vous pouvez augmenter la taille du volume de vidage jusqu'à 40 Go sur un système contenant une quantité de mémoire importante comme suit :

```
# zfs set volsize=40G rpool/dump
```

Le redimensionnement d'un volume de vidage de grande ampleur peut prendre un certain temps.

Si, pour une raison quelconque, vous devez activer un périphérique de vidage après l'avoir créé manuellement, utilisez une syntaxe semblable à la suivante :

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

- Un système avec 128 Go de mémoire ou plus nécessite un périphérique de vidage plus important que celui créé par défaut. Si le périphérique de vidage est trop petit pour capturer un vidage mémoire sur incident existant, un message semblable au suivant s'affiche :

```
# dumpadm -d /dev/zvol/dsk/rpool/dump
dumpadm: dump device /dev/zvol/dsk/rpool/dump is too small to hold a system dump
dump size 36255432704 bytes, device size 34359738368 bytes
```

Pour plus d'informations sur la taille des périphériques de swap et de vidage, reportez-vous à la section [“Planning for Swap Space”](#) du *System Administration Guide: Devices and File Systems*.

- Vous ne pouvez pas ajouter actuellement un périphérique de vidage à un pool avec plusieurs périphériques de niveau supérieur. Un message similaire à celui figurant ci-dessous s'affiche :

```
# dumpadm -d /dev/zvol/dsk/datapool/dump
dump is not supported on device '/dev/zvol/dsk/datapool/dump': 'datapool' has multiple top level vdevs
```

Ajoutez le périphérique de vidage au pool racine. Ce dernier ne peut pas contenir plusieurs périphériques de niveau supérieur.

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 de l'archive d'initialisation et du fichier noyau sont résolus dans le système de fichiers racine sélectionné pour l'initialisation.

Lorsque vous réinitialisez un système en vue d'une installation, un disque RAM est utilisé pour le système de fichiers racine tout au long du processus d'installation.

L'initialisation à partir d'un système de fichiers ZFS diffère de celle à partir d'un système de fichiers UFS car avec ZFS, un spécificateur de périphérique d'initialisation identifie un pool de stockage par opposition à un seul système de fichiers 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 identifié par la propriété `boot fs` du pool. Cette sélection par défaut peut être remplacée en spécifiant un jeu de données d'initialisation alternatif 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. Pour de plus amples informations, consultez les références suivantes :

- [“Installation d'un système de fichiers racine ZFS \(installation initiale\)”](#) à la page 130
- [“Création d'un pool racine mis en miroir \(post-installation\)”](#) à la page 136

Consultez les problèmes connus suivants relatifs aux pools racine ZFS mis en miroir :

- 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` pour 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 syntaxe de la commande `installboot` ou la commande `installgrub` serait comme suit :
 - SPARC :

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

- x86 :

```
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) du pool suivant.

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

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

- SPARC : entrez le disque de rechange à l'invite ok.

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

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

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

- x86 : sélectionnez un autre disque dans le pool ZFS racine mis en miroir à partir du menu BIOS correspondant.

Utilisez ensuite une syntaxe semblable à la syntaxe suivante pour confirmer l'initialisation à partir du disque de rechange :

```
x86# prtconf -v|sed -n '/bootpath/,/value/p'
name='bootpath' type=string items=1
value='/pci@0,0/pci@086,25f8@4/pci@08e,286@0/disk@0,0:a'
```

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

Sur un système SPARC avec environnements d'initialisation ZFS multiples, vous pouvez initialiser à partir de tout environnement d'initialisation en utilisant la commande `luactivate`.

Au cours de l'installation du SE Solaris et de la procédure Oracle 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 zfsBE
bootfs rpool/ROOT/zfsBE
title zfs2BE
bootfs rpool/ROOT/zfs2BE
```

Lorsqu'un nouvel environnement d'initialisation est créé, le fichier `menu.lst` est mis à jour automatiquement.

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

- Une fois l'environnement d'initialisation activé, vous pouvez utiliser la commande d'initialisation `-L` pour afficher une liste de jeux de données 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.
- Vous pouvez utiliser la commande `-Z dataset` pour initialiser un jeu de données ZFS spécifique.

EXEMPLE 5-8 SPARC : Initialisation à partir d'un environnement d'initialisation ZFS spécifique

Si vous disposez de plusieurs environnements d'initialisation ZFS dans un pool de stockage ZFS situé sur le périphérique d'initialisation de votre système, vous pouvez utiliser la commande `luactivate` pour spécifier un environnement d'initialisation par défaut.

Par exemple, 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
-----
zfsBE                  yes     no     no       yes   -
zfs2BE                  yes     yes    yes      no    -
```

Si votre système SPARC comporte plusieurs environnements d'initialisation ZFS, vous pouvez également utiliser la commande `boot -L` pour initialiser le système à partir d'un autre environnement d'initialisation que celui par défaut. Cependant, un environnement d'initialisation démarré à partir d'une session `boot -L` n'est pas redéfini en tant qu'environnement d'initialisation par défaut et la propriété `bootfs` n'est pas mise à jour. Si vous souhaitez qu'un environnement d'initialisation démarre par défaut à partir d'une session `boot -L`, vous devez activer ce dernier avec la commande `luactivate`.

EXEMPLE 5-8 SPARC : Initialisation à partir d'un environnement d'initialisation ZFS spécifique (Suite)

Exemple :

```
ok boot -L
Rebooting with command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0 File and args: -L

1 zfsBE
2 zfs2BE
Select environment to boot: [ 1 - 2 ]: 1
To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfsBE

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

EXEMPLE 5-9 SPARC : Initialisation d'un système de fichiers ZFS en mode de secours

Vous pouvez initialiser un système SPARC à partir de l'archive de secours située dans `/platform/`uname -i`/failsafe`, comme suit .

```
ok boot -F failsafe
```

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 rpool/ROOT/zfsBE -F failsafe
```

x86 : initialisation à partir d'un système de fichiers ZFS racine

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

```
title Solaris 10 9/10 X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Si 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 5-10 x86 : Initialisation d'un système de fichiers ZFS

Lorsqu'un système s'initialise 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 de paramètre, tout comme tous les paramètres spécifiés par l'option `-B`, est transmise par GRUB au noyau. Exemple :

```
title Solaris 10 9/10 X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

EXEMPLE 5-11 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
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Résolution de problèmes de point de montage empêchant l'initialisation (Solaris 10 10/08)

Le meilleur moyen de changer 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 à partir d'un environnement différent consiste à le sélectionner lors de l'initialisation. Vous pouvez sélectionner un autre environnement d'initialisation à partir du menu GRUB sur un système x86 ou en effectuant une initialisation explicite à partir du PROM sur un système SPARC.

Étant donné la présence d'un bogue dans la fonctionnalité Oracle Solaris Live Upgrade de Solaris 10 10/08, il se peut que l'environnement d'initialisation inactif ne puisse pas s'initialiser. Ce problème est lié à la présence d'un point de montage non valide dans les jeux de données ZFS

ou dans le jeu de données ZFS 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.

Si le jeu de données d'une zone a un point de montage non valide, le point de montage peut être corrigé en procédant comme suit.

▼ Résolution des problèmes de point de montage ZFS

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

2 Importez le pool.

Exemple :

```
# zpool import rpool
```

3 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 pouvez 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.

Initialisation en vue d'une récupération d'un environnement racine ZFS

Suivez la procédure suivante si vous devez initialiser le système pour pouvoir récupérer un mot de passe racine perdu ou tout problème similaire.

Vous devez démarrer en mode de secours ou en mode d'initialisation à partir d'autres média en fonction de la gravité de l'erreur. En règle générale, vous pouvez effectuer une initialisation en mode de secours pour récupérer un mot de passe racine perdu ou inconnu.

- “Initialisation d'un système de fichiers ZFS en mode de secours” à la page 177
- “Initialisation d'un système de fichiers ZFS à partir d'un autre média” à la page 178

Si vous devez récupérer un pool racine ou un instantané de pool racine, reportez-vous à la section “Restauration du pool racine ZFS ou des instantanés du pool racine” à la page 178.

▼ Initialisation d'un système de fichiers ZFS en mode de secours

1 Effectuez une initialisation en mode de secours.

Sur un système SPARC :

```
ok boot -F failsafe
```

Pour les systèmes x86, sélectionnez le mode de secours à partir de l'invite GRUB.

2 Lorsque vous y êtes invité, montez l'environnement d'exécution ZFS sur le répertoire /a :

```
.
.
.
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
Starting shell.
```

3 Accédez au répertoire /a/etc.

```
# cd /a/etc
```

4 Définissez le type TERM, le cas échéant.

```
# TERM=vt100
# export TERM
```

5 Corrigez le fichier passwd ou shadow.

```
# vi shadow
```

6 Redémarrez le système.

```
# init 6
```

▼ Initialisation d'un système de fichiers ZFS à partir d'un autre média

Si un problème empêche le système de s'initialiser correctement ou si tout autre problème grave se produit, vous devez effectuer une initialisation à partir d'un serveur d'installation sur le réseau ou à partir d'un CD-ROM d'installation Solaris, importer le pool racine, monter l'environnement d'initialisation ZFS et tenter de résoudre le problème.

1 Initialisez le système à partir d'un CD d'installation ou à partir du réseau.

- SPARC :

```
ok boot cdrom -s
ok boot net -s
```

Si vous n'utilisez pas l'option -s, vous devrez quitter le programme d'installation.

- x86 : initialisez le système à partir d'un réseau ou d'un CD en local.

2 Importez le pool racine et spécifiez un autre point de montage. Exemple :

```
# zpool import -R /a rpool
```

3 Montez l'environnement d'initialisation ZFS. Exemple :

```
# zfs mount rpool/ROOT/zfsBE
```

4 Accédez à l'environnement d'initialisation ZFS à partir du répertoire /a.

```
# cd /a
```

5 Redémarrez le système.

```
# init 6
```

Restauration du pool racine ZFS ou des instantanés du pool racine

Les sections suivantes décrivent comment effectuer les tâches ci-dessous :

- “Remplacement d'un disque dans le pool racine ZFS” à la page 178
- “Création d'instantanés de pool racine” à la page 181
- “Recréation d'un pool racine ZFS et restauration d'instantanés de pool racine” à la page 182
- “Restauration des instantanés d'un pool racine à partir d'une initialisation de secours ” à la page 184

▼ Remplacement d'un disque dans le pool racine ZFS

Vous pouvez être amené à remplacer un disque dans le pool racine pour les raisons suivantes :

- Le pool racine est trop petit et vous souhaitez remplacer un disque plus petit par un disque plus grand.
- Un pool de disques est défectueux. Dans un pool non redondant, si le disque est défectueux et empêche l'initialisation du système, vous devrez initialiser votre système à partir d'un autre support, par exemple un CD ou le réseau, avant de remplacer le disque du pool racine.

Dans le cadre d'une configuration de pool racine mise en miroir, vous pouvez tenter de remplacer le disque sans effectuer une initialisation à partir d'un autre média. Vous pouvez remplacer un disque défaillant en utilisant la commande `zpool replace`. Si vous disposez d'un autre disque, vous pouvez également utiliser la commande `zpool attach`. Pour savoir comment connecter un autre disque et déconnecter un disque de pool racine, reportez-vous à la procédure de cette section.

Avec certains composants matériels, vous devez déconnecter le disque et en supprimer la configuration avant de tenter d'utiliser la commande `zpool replace` pour remplacer le disque défectueux. Exemple :

```
# zpool offline rpool c1t0d0s0
# cfgadm -c unconfigure cl::dsk/c1t0d0
<Physically remove failed disk c1t0d0>
<Physically insert replacement disk c1t0d0>
# cfgadm -c configure cl::dsk/c1t0d0
# zpool replace rpool c1t0d0s0
# zpool online rpool c1t0d0s0
# zpool status rpool
<Let disk resilver before installing the boot blocks>
SPARC# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t0d0s0
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t9d0s0
```

Avec certains composants matériels, il n'est pas nécessaire de connecter le disque, ni de reconfigurer son remplacement après son insertion.

Vous devez identifier les chemins d'accès du périphérique d'initialisation du nouveau disque et du disque actuel afin de tester l'initialisation à partir du disque de remplacement et afin de pouvoir initialiser manuellement le système à partir du disque existant, en cas de dysfonctionnement du disque de remplacement. Dans l'exemple de la procédure suivante, le nom du chemin du disque de pool racine actuel (`c1t10d0s0`) est le suivant :

```
/pci@8,700000/pci@3/scsi@5/sd@a,0
```

Le nom du chemin du disque d'initialisation de remplacement (`c1t9d0s0`) est le suivant :

```
/pci@8,700000/pci@3/scsi@5/sd@9,0
```

1 Connectez physiquement le disque de remplacement (nouveau disque).

2 Confirmez que le nouveau disque possède une étiquette SMI et une tranche 0.

Pour plus d'informations sur le nouvel étiquetage d'un disque destiné au pool racine, visitez le site suivant :

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

3 Associez le nouveau disque au pool racine.

Exemple :

```
# zpool attach rpool c1t10d0s0 c1t9d0s0
```

4 Confirmez le statut du pool racine.

Exemple :

```
# zpool status rpool
pool: rpool
state: ONLINE
status: One or more devices is currently being resilvered. The pool will
continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scrub: resilver in progress, 25.47% done, 0h4m to go
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c1t10d0s0	ONLINE	0	0	0
c1t9d0s0	ONLINE	0	0	0

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

5 Une fois la resynchronisation terminée, appliquez les blocs d'initialisation au nouveau disque.

Utilisez une syntaxe semblable à la syntaxe suivante :

- SPARC :

```
# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t9d0s0
```

- x86 :

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t9d0s0
```

6 Vérifiez que vous pouvez initialiser le système à partir du nouveau disque.

Pour les systèmes SPARC, vous devez par exemple utiliser une syntaxe semblable à la syntaxe suivante :

```
ok boot /pci@8,700000/pci@3/scsi@5/sd@9,0
```

7 Si le système s'initialise à partir du nouveau disque, déconnectez l'ancien disque.

Exemple :

```
# zpool detach rpool c1t10d0s0
```

- 8 **Configurez le système pour qu'il puisse s'initialiser automatiquement à partir du nouveau disque, à l'aide de la commande `eeeprom` ou de la commande `setenv` de la PROM d'initialisation SPARC, ou reconfigurez le BIOS du PC.**

▼ Création d'instantanés de pool racine

Vous pouvez créer des instantanés de pool racine à des fins de récupération. La meilleure façon de créer des instantanés de pool racine consiste à effectuer un instantané récursif du pool racine.

La procédure suivante crée un instantané de pool racine récursif et le stocke en tant que fichier dans un pool sur un système distant. En cas de défaillance du pool racine, le jeu de données distant peut être monté à l'aide de NFS et le fichier d'instantané peut être reçu dans le pool recréé. Vous pouvez également stocker les instantanés de pool racine en tant qu'instantanés réels dans un pool d'un système distant. L'envoi et la réception des instantanés à partir d'un système distant est un peu plus complexe, car vous devez configurer `ssh` ou utiliser `rsh` pendant que le système à réparer est initialisé à partir de la mini-racine du système d'exploitation Solaris.

Pour plus d'informations sur le stockage et la récupération d'instantanés de pool racine à distance et afin d'obtenir les informations les plus récentes sur la récupération de pool racine, visitez le site :

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

La validation des instantanés stockés à distance en tant que fichiers ou instantanés est une étape importante dans la récupération du pool racine. En appliquant l'une des deux méthodes, les instantanés doivent être recréés régulièrement, par exemple, lorsque la configuration du pool est modifiée ou lorsque le SE Solaris est mis à niveau.

Dans la procédure suivante, le système est initialisé à partir de l'environnement d'initialisation `zfsBE`.

- 1 **Créez un pool et un système de fichiers sur un système distant pour stocker les instantanés.**

Exemple :

```
remote# zfs create rpool/snaps
```

- 2 **Partagez le système de fichiers avec le système local.**

Exemple :

```
remote# zfs set sharenfs='rw=local-system,root=local-system' rpool/snaps
# share
-@rpool/snaps /rpool/snaps sec=sys,rw=local-system,root=local-system ""
```

- 3 **Créez un instantané récursif du pool racine.**

```
local# zfs snapshot -r rpool@0804
local# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	6.17G	60.8G	98K	/rpool
rpool@0804	0	-	98K	-
rpool/ROOT	4.67G	60.8G	21K	/rpool/ROOT
rpool/ROOT@0804	0	-	21K	-
rpool/ROOT/zfsBE	4.67G	60.8G	4.67G	/
rpool/ROOT/zfsBE@0804	386K	-	4.67G	-
rpool/dump	1.00G	60.8G	1.00G	-
rpool/dump@0804	0	-	1.00G	-
rpool/swap	517M	61.3G	16K	-
rpool/swap@0804	0	-	16K	-

4 Envoyez les instantanés du pool racine au système distant.

Exemple :

```
local# zfs send -Rv rpool@0804 > /net/remote-system/rpool/snaps/rpool.0804
sending from @ to rpool@0804
sending from @ to rpool/swap@0804
sending from @ to rpool/ROOT@0804
sending from @ to rpool/ROOT/zfsBE@0804
sending from @ to rpool/dump@0804
```

▼ Recréation d'un pool racine ZFS et restauration d'instantanés de pool racine

Dans cette procédure, on suppose les conditions suivantes :

- Le pool racine ZFS ne peut pas être récupéré.
- Les instantanés du pool racine ZFS sont stockés sur un système distant et sont partagés sur NFS.

Toutes les étapes sont effectuées sur le système local.

1 Initialisez le système à partir d'un CD/DVD ou du réseau.

- SPARC : sélectionnez l'une des méthodes d'initialisation suivantes :

```
ok boot net -s
ok boot cdrom -s
```

Si vous n'utilisez pas l'option -s, vous devrez quitter le programme d'installation.

- x86 : initialisez le système à partir du DVD ou du réseau. Quittez ensuite le programme d'installation.

2 Montez le jeu de données de l'instantané distant.

Exemple :

```
# mount -F nfs remote-system:/rpool/snaps /mnt
```

Si vos services réseau ne sont pas configurés, il peut être nécessaire de spécifier l'adresse IP du *système distant*.

3 Si le disque du pool racine est remplacé et ne contient aucune étiquette de disque pouvant être utilisée par ZFS, vous devez renommer le disque.

Pour en savoir plus sur l'affectation d'un nouveau nom au disque, reportez-vous au site suivant :

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

4 Recréez le pool racine.

Exemple :

```
# zpool create -f -o failmode=continue -R /a -m legacy -o cachefile=
/etc/zfs/zpool.cache rpool c1t1d0s0
```

5 Restaurez les instantanés du pool racine.

Cette étape peut prendre un certain temps. Exemple :

```
# cat /mnt/rpool.0804 | zfs receive -Fdu rpool
```

L'utilisation de l'option `-u` implique que l'archive restaurée n'est pas montée à la fin de l'opération `zfs receive`.

6 Vérifiez que les jeux de données du pool racine sont restaurés.

Exemple :

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                6.17G  60.8G   98K    /a/rpool
rpool@0804           0      -       98K    -
rpool/ROOT           4.67G  60.8G   21K    /legacy
rpool/ROOT@0804     0      -       21K    -
rpool/ROOT/zfsBE    4.67G  60.8G  4.67G  /a
rpool/ROOT/zfsBE@0804 398K   -       4.67G  -
rpool/dump          1.00G  60.8G  1.00G  -
rpool/dump@0804     0      -       1.00G  -
rpool/swap          517M   61.3G   16K    -
rpool/swap@0804     0      -       16K    -
```

7 Définissez la propriété `bootfs` sur l'environnement d'initialisation du pool racine.

Exemple :

```
# zpool set bootfs=rpool/ROOT/zfsBE rpool
```

8 Installez les blocs d'initialisation sur le nouveau disque.

SPARC :

```
# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t1d0s0
```

x86 :

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

9 Redémarrez le système.

```
# init 6
```

▼ Restauration des instantanés d'un pool racine à partir d'une initialisation de secours

Cette procédure part du principe que les instantanés du pool racine existant sont disponibles. Dans l'exemple suivant, ils se trouvent sur le système local.

```
# zfs snapshot -r rpool@0804
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                6.17G 60.8G   98K    /rpool
rpool@0804                            0      -     98K    -
rpool/ROOT                            4.67G 60.8G   21K    /rpool/ROOT
rpool/ROOT@0804                        0      -     21K    -
rpool/ROOT/zfsBE                      4.67G 60.8G  4.67G    /
rpool/ROOT/zfsBE@0804                 398K   -   4.67G    -
rpool/dump                            1.00G 60.8G  1.00G    -
rpool/dump@0804                        0      -   1.00G    -
rpool/swap                            517M  61.3G   16K    -
rpool/swap@0804                        0      -     16K    -
```

1 Arrêtez le système et initialisez-le en mode secours.

```
ok boot -F failsafe
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
```

```
Starting shell.
```

2 Annulez (roll back) chaque instantané du pool racine.

```
# zfs rollback rpool@0804
# zfs rollback rpool/ROOT@0804
# zfs rollback rpool/ROOT/zfsBE@0804
```

3 Réinitialisez le système en mode multi-utilisateur.

```
# init 6
```


Gestion des systèmes de fichiers Oracle Solaris ZFS

Ce chapitre contient des informations détaillées sur la gestion des systèmes de fichiers ZFS Oracle Solaris. Il aborde notamment les concepts d'organisation hiérarchique des systèmes de fichiers, d'héritage des propriétés, de gestion automatique des points de montage et d'interaction sur les partages.

Il contient les sections suivantes :

- “Gestion des systèmes de fichiers ZFS (présentation)” à la page 185
- “Création, destruction et renommage de systèmes de fichiers ZFS” à la page 186
- “Présentation des propriétés ZFS” à la page 189
- “Envoi de requêtes sur les informations des systèmes de fichiers ZFS” à la page 204
- “Gestion des propriétés ZFS” à la page 206
- “Montage et partage des systèmes de fichiers ZFS” à la page 211
- “Définition des quotas et réservations ZFS” à la page 218

Gestion des systèmes de fichiers ZFS (présentation)

La création d'un système de fichiers ZFS s'effectue sur un pool de stockage. La création et la destruction des systèmes de fichiers peuvent s'effectuer de manière dynamique, sans allocation ni formatage manuels de l'espace disque sous-jacent. En raison de leur légèreté et de leur rôle central dans l'administration du système ZFS, la création de ces systèmes de fichiers constitue généralement une opération extrêmement courante.

La gestion des systèmes de fichiers ZFS s'effectue à l'aide de la commande `zfs`. La commande `zfs` offre un ensemble de sous-commandes permettant d'effectuer des opérations spécifiques sur les systèmes de fichiers. Chacune de ces sous-commandes est décrite en détail dans ce chapitre. Cette commande permet également de gérer les instantanés, les volumes et les clones. Toutefois, ces fonctionnalités sont uniquement traitées de manière succincte dans ce chapitre. Pour plus d'informations sur les instantanés et les clones, reportez-vous au [Chapitre 7](#), “Utilisation des instantanés et des clones ZFS Oracle Solaris”. Pour de plus amples informations sur les volumes ZFS, reportez-vous à la section “Volumes ZFS” à la page 281.

Remarque – Dans ce chapitre, le terme *jeu de données* désigne de manière générique un système de fichiers, un instantané, un clone ou un volume.

Création, destruction et renommage de systèmes de fichiers ZFS

La création et la destruction des systèmes de fichiers ZFS s'effectuent respectivement à l'aide des commandes `zfs create` et `zfs destroy`. Vous pouvez renommer les systèmes de fichiers ZFS en utilisant la commande `zfs rename`.

- “Création d'un système de fichiers ZFS” à la page 186
- “Destruction d'un système de fichiers ZFS” à la page 187
- “Modification du nom d'un système de fichiers ZFS” à la page 188

Création d'un système de fichiers ZFS

La création des systèmes de fichiers ZFS s'effectue à l'aide de la commande `zfs create`. La sous-commande `create` ne peut contenir qu'un argument : le nom du système de fichiers à créer. Le nom de ce système de fichiers permet également de définir le nom du chemin par rapport au nom du pool, comme suit :

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. 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 52.

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 système de fichiers `bonwick` créé 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 211.

Pour plus d'informations sur la commande `zfs create`, reportez-vous à la page de manuel `zfs(1M)`.

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

Dans l'exemple ci-dessous, le point de montage `/export/zfs` est créé pour le système de fichiers `tank/home` :

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

Pour plus d'informations sur les propriétés des systèmes de fichiers, reportez-vous à la section [“Présentation des propriétés ZFS” à la page 189](#).

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 212](#).

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 descendants. Pour détruire un système de fichiers et l'ensemble des descendants de ce système de fichiers, indiquez l'option `-r`. Ce type d'opération de destruction récursive entraîne également la destruction des instantanés ; l'utilisation de cette option requiert donc une attention particulière.

```
# zfs destroy tank/ws
cannot destroy 'tank/ws': filesystem has children
use '-r' to destroy the following datasets:
tank/ws/billm
```

```
tank/ws/bonwick
tank/ws/maybee
```

```
# zfs destroy -r tank/ws
```

Si le système de fichiers à détruire possède des systèmes indirectement dépendants, même la commande de destruction récursive échoue. Pour forcer la destruction de *tous* 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` avec la commande `zfs destroy`. L'utilisation de ces options requiert donc une attention particulière.

Pour plus d'informations sur les instantanés et les clones, reportez-vous au [Chapitre 7](#), "Utilisation des instantanés et des clones ZFS Oracle Solaris".

Modification du nom d'un système de fichiers ZFS

La modification du nom d'un système de fichiers ZFS s'effectue à l'aide de la commande `zfs rename`. La commande `rename` permet d'effectuer les opérations suivantes :

- Modifier le nom d'un système de fichiers.
- Déplacer le système de fichiers au sein de la hiérarchie ZFS.
- Modifier le nom d'un système de fichiers et son emplacement au sein de la hiérarchie ZFS.

L'exemple suivant utilise la sous-commande `rename` pour renommer le système de fichiers `kustarz` en `kustarz_old` :

```
# zfs rename tank/home/kustarz tank/home/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 disque disponible doit

être suffisant pour contenir le nouveau système de fichiers. Si l'espace disque disponible du nouvel emplacement est insuffisant, par exemple si le quota d'espace disque est atteint, l'opération `rename` échoue.

Pour plus d'informations sur les quotas, reportez-vous à la section [“Définition des quotas et réservations ZFS” à la page 218](#).

L'opération `rename` tente de démonter, puis de remonter le système de fichiers ainsi que ses éventuels systèmes de fichiers descendants. Si la commande `rename` ne parvient pas à démonter un système de fichiers actif, l'opération échoue. Si ce problème survient, vous devez forcer le démontage du système de fichiers.

Pour plus d'informations sur la modification du nom des instantanés, reportez-vous à la section [“Renommage d'instantanés ZFS” à la page 228](#).

Présentation des propriétés ZFS

Les propriétés constituent le mécanisme principal de contrôle du comportement des systèmes de fichiers, des volumes, des instantanés et des clones. Sauf mention contraire, les propriétés définies dans cette section s'appliquent à tous les types de jeu de données.

- [“Propriétés ZFS natives en lecture seule” à la page 198](#)
- [“Propriétés ZFS natives définies” à la page 199](#)
- [“Propriétés ZFS définies par l'utilisateur” à la page 202](#)

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 202](#).

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 jeu de données parent, peuvent être appliquées à l'ensemble des descendants de ce parent.

Toutes ces propriétés héritables sont associées à une source indiquant la façon dont la propriété a été obtenue. Les sources de propriétés peuvent être définies sur les valeurs suivantes :

`local`

Indique que la propriété 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 206](#).

<code>inherited from</code> <i>nom-jeu-données</i>	Indique que la propriété a été héritée à partir de l'ascendant indiqué.
<code>default</code>	Indique que la valeur de la propriété n'a été ni héritée, ni définie en local. Cette source est définie lorsque la propriété n'est pas définie en tant que source <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 202.

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 “Propriétés ACL” à la page 247.
<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 “Propriétés ACL” à la page 247.
<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 disque disponible pour le jeu de données et l'ensemble des systèmes enfant, sans tenir compte des autres activités du pool. L'espace disque étant partagé au sein d'un pool, l'espace disponible peut être limité par divers facteurs, y compris la taille du pool physique, les quotas, les réservations ou les autres jeux de données présents au sein du pool.</p> <p>L'abréviation de la propriété est avail.</p> <p>Pour plus d'informations sur la détermination de l'espace disque, reportez-vous à la section "Comptabilisation de l'espace disque ZFS" à la page 62.</p>
canmount	Booléen	on	<p>Détermine si un système de fichiers donné 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 <code>canmount</code>" à la page 201.</p>
checksum	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>fletcher4</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>	Active ou désactive la compression d'un 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> . L'activation de la compression sur un système de fichiers contenant des données existantes entraîne uniquement la compression des nouvelles données. Les données actuelles restent non compressées. L'abréviation de la propriété est <code>compress</code> .
<code>compressratio</code>	Valeur numérique	SO	Propriété en lecture seule indiquant le ratio de compression obtenu pour un jeu de données, exprimé sous la forme d'un multiple. La compression peut être activée en exécutant <code>zfs set compression=on dataset</code> . Cette valeur est calculée sur la base de la taille logique de l'ensemble des fichiers et de la quantité de données physiques indiquée. Elle induit un gain explicite basé sur l'utilisation de la propriété <code>compression</code> .
<code>copies</code>	Valeur numérique	1	Définit le nombre de copies des données utilisateur par système de fichiers. Les valeurs disponibles sont 1, 2 ou 3. Ces copies viennent s'ajouter à toute redondance au niveau du pool. L'espace disque utilisé par plusieurs copies de données utilisateur est chargé dans le fichier et le jeu de données correspondants et pénalise les quotas et les réservations. En outre, la propriété <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.
<code>creation</code>	Chaîne	SO	Propriété en lecture seule identifiant la date et l'heure de création d'un jeu de données.
<code>devices</code>	Booléen	<code>on</code>	Contrôle si les fichiers de périphérique d'un système de fichiers peuvent être ouverts.
<code>exec</code>	Booléen	<code>on</code>	Contrôle l'autorisation d'exécuter les programmes dans un 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.

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 un système de fichiers, un clone ou un instantané est actuellement monté. Cette propriété ne s'applique pas aux volumes. Les valeurs possibles sont yes ou no.
mountpoint	Chaîne	SO	Détermine le point de montage utilisé pour le système de fichiers. Lorsque la propriété mountpoint d'un système de fichiers est modifiée, ce système de fichiers ainsi que les éventuels systèmes descendants héritant du point de montage sont démontés. Si la nouvelle valeur est définie sur legacy, ces systèmes restent démontés. Dans le cas contraire, ils sont automatiquement remontés au nouvel emplacement si la propriété était précédemment définie sur legacy ou sur none ou s'ils étaient montés avant la modification de la propriété. D'autre part, le partage de tout système de fichiers est annulé puis rétabli au nouvel emplacement. Pour plus d'informations sur l'utilisation de cette propriété, reportez-vous à la section “Gestion des points de montage ZFS” à la page 211.
primarycache	Chaîne	all	Contrôle les éléments mis en cache dans le cache principal (ARC). Les valeurs possibles sont all, none et metadata. Si elles sont définies sur all, les données d'utilisateur et les métadonnées sont mises en cache. Si elle est définie sur none, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si ces valeurs sont définies sur metadata, seules les métadonnées sont mis en mémoire cache.
origin	Chaîne	SO	Propriété en lecture seule appliquée aux systèmes de fichiers ou aux volumes clonés et indiquant l'instantané à partir duquel le clone a été créé. Le système d'origine ne peut pas être détruit (même à l'aide des options -r ou -f) tant que le clone existe. Les systèmes de fichiers non clonés indiquent un système d'origine none.

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
quota	Valeur numérique (ou none)	none	<p>Limite la quantité d'espace disque disponible pour le jeu de données et ses descendants. Cette propriété permet d'appliquer une limite fixe à la quantité d'espace disque utilisée, y compris l'espace utilisé par les descendants, qu'il s'agisse de systèmes de fichiers ou d'instantanés. La définition d'un quota sur un descendant 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.</p> <p>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 219.</p>
readonly	Booléen	off	<p>Contrôle l'autorisation de modifier un jeu de données. Lorsqu'elle est définie sur on, aucune modification ne peut être apportée.</p> <p>L'abréviation de la propriété est <code>rdonly</code>.</p>
recordsize	Valeur numérique	128K	<p>Spécifie une taille de bloc suggérée pour les fichiers d'un système de fichiers.</p> <p>L'abréviation de la propriété est <code>recsize</code>. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété recordsize” à la page 201.</p>
referenced	Valeur numérique	SO	<p>Propriété en lecture seule identifiant la quantité de données à laquelle un 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 disque que le système de fichiers ou l'instantané à partir duquel il a été créé. En effet, son contenu est identique.</p> <p>L'abréviation de la propriété est <code>refer</code>.</p>
refquota	Valeur numérique (ou none)	none	<p>Définit la quantité d'espace disque pouvant être utilisé par un jeu de données. Cette propriété définit une quantité d'espace maximale. Cette limite maximale n'inclut pas l'espace disque utilisé par les descendants, notamment les instantanés et les clones.</p>

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
<code>refreservation</code>	Valeur numérique (ou none)	none	<p>Définit la quantité d'espace disque minimale garantie pour un jeu de données, à l'exclusion des descendants, notamment les instantanés et les clones. Lorsque la quantité d'espace disque utilisée est inférieure à cette valeur, le système considère que le jeu de données utilise la quantité d'espace spécifiée par <code>refreservation</code>. La réservation <code>refreservation</code> est prise en compte dans l'espace disque 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>L'abréviation de la propriété est <code>refserv</code>.</p>
<code>reservation</code>	Valeur numérique (ou none)	none	<p>Définit la quantité minimale d'espace disque réservée pour un jeu de données et ses descendants. Lorsque la quantité d'espace disque 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 disque utilisé des jeux de données parent et viennent en déduction de leurs quotas et réservations.</p> <p>L'abréviation de la propriété est <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 222.</p>
<code>secondarycache</code>	Chaîne	<code>all</code>	<p>Contrôle les éléments qui sont mis en cache dans le cache secondaire (L2ARC). Les valeurs possibles sont <code>all</code>, <code>none</code> et <code>metadata</code>. Si elles sont définies sur <code>all</code>, les données d'utilisateur et les métadonnées sont mises en cache. Si elle est définie sur <code>none</code>, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si ces valeurs sont définies sur <code>metadata</code>, seules les métadonnées sont mis en mémoire cache.</p>
<code>setuid</code>	Booléen	<code>on</code>	<p>Contrôle l'application du bit <code>setuid</code> dans un système de fichiers.</p>

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
<code>shareiscsi</code>	Chaîne	<code>off</code>	Contrôle si un volume ZFS est partagé en tant que cible iSCSI. Les valeurs de la propriété sont <code>on</code> , <code>off</code> et <code>type=disk</code> . Vous pouvez définir la valeur <code>shareiscsi=on</code> pour un système de fichiers afin que tous les volumes ZFS au sein du système de fichiers soient partagés par défaut. Cependant, la configuration de cette propriété sur un système de fichiers n'a aucune incidence directe.
<code>sharenfs</code>	Chaîne	<code>off</code>	Détermine si un 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 file</code> . 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 216.
<code>snapdir</code>	Chaîne	<code>hidden</code>	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 225.
<code>type</code>	Chaîne	<code>SO</code>	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>).
<code>used</code>	Valeur numérique	<code>SO</code>	Propriété en lecture seule identifiant la quantité d'espace disque utilisée par le jeu de données et tous ses descendants. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété used” à la page 199.
<code>usedbychildren</code>	Valeur numérique	<code>off</code>	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par les enfants de ce jeu de données, qui serait libérée si tous ses enfants étaient détruits. L'abréviation de la propriété est <code>usedchild</code> .

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
<code>usedbydataset</code>	Valeur numérique	<code>off</code>	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par le jeu de données lui-même, qui serait libérée si ce dernier était détruit, après la destruction préalable de tous les instantanés et la suppression de toutes les réservations <code>reservation</code> . L'abréviation de la propriété est <code>usedds</code> .
<code>usedbyreservation</code>	Valeur numérique	<code>off</code>	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par un jeu <code>reservation</code> sur un jeu de données, qui serait libérée si le jeu <code>reservation</code> était supprimé. L'abréviation de la propriété est <code>usedrefreserv</code> .
<code>usedbysnapshots</code>	Valeur numérique	<code>off</code>	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par les instantanés de ce jeu de données. En particulier, elle correspond à la quantité d'espace disque qui serait libérée si l'ensemble des instantanés de ce jeu de données était supprimé. Notez que cette valeur ne correspond pas simplement à la somme des propriétés <code>used</code> des instantanés, car l'espace peut être partagé par plusieurs instantanés. L'abréviation de la propriété est <code>usedsnap</code> .
<code>version</code>	Valeur numérique	SO	Identifie la version du disque d'un système de fichiers. Cette information n'est pas liée à la version du pool. Cette propriété peut uniquement être définie avec une version supérieure prise en charge par la version du logiciel. Pour plus d'informations, reportez-vous à la commande <code>zfs upgrade</code> .
<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 202.
<code>volblocksize</code>	Valeur numérique	8 KB	(Volumes) Spécifie la taille de bloc du volume. Une fois que des données ont été écrites sur un volume, la taille de bloc ne peut plus être modifiée. Vous devez donc définir cette valeur lors de la création du volume. La taille de bloc par défaut des volumes est de 8 Ko. Toute puissance de deux comprise entre 512 octets et 128 Ko est correcte. L'abréviation de la propriété est <code>volblock</code> .

TABLEAU 6-1 Description des propriétés ZFS natives (Suite)

Nom de la propriété	Type	Valeur par défaut	Description
zoned	Booléen	SO	Indique si un 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 284.
xattr	Booléen	on	Indique si les attributs étendus sont activés (on) ou désactivés (off) pour ce système de fichiers.

Propriétés ZFS natives en lecture seule

Les propriétés natives en lecture seule peuvent être récupérées, mais ne peuvent pas être définies. Elles ne peuvent pas non plus être héritées. Certaines propriétés natives sont spécifiques à un type de jeu de données. Dans ce cas, le type de jeu de données 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
- compressratio
- creation
- mounted
- origin
- referenced
- type
- used

Pour plus d'informations sur cette propriété, reportez-vous à la section "Propriété used" à la page 199.

- usedbychildren
- usedbydataset
- usedbyreservation
- usedbysnapshots

Pour plus d'informations sur la détermination de l'espace disque, notamment sur les propriétés `used`, `referenced` et `available`, reportez-vous à la section [“Comptabilisation de l'espace disque ZFS”](#) à la page 62.

Propriété `used`

La propriété `used` est une propriété en lecture seule indiquant la quantité d'espace disque utilisée par le jeu de données et l'ensemble de ses descendants. Cette valeur est comparée au quota et à la réservation définis pour le jeu de données. La quantité d'espace disque utilisé n'inclut pas la réservation du jeu de données. En revanche, elle prend en compte les réservations définies pour les éventuels jeux de données descendants. La quantité d'espace disque utilisée sur le parent par un jeu de données, ainsi que la quantité d'espace disque libérée si le jeu de données est détruit de façon récursive, constituent la plus grande partie de son espace utilisé et sa réservation.

Lors de la création d'un instantané, l'espace disque correspondant est dans un premier temps partagé entre cet instantané et le système de fichiers ainsi que les instantanés existants (le cas échéant). Lorsque le système de fichiers est modifié, l'espace disque précédemment partagé devient dédié à l'instantané. Il est alors comptabilisé dans l'espace utilisé par cet instantané. L'espace disque utilisé par un instantané représente ses données uniques. La suppression d'instantanés peut également augmenter l'espace disque dédié et utilisé par les autres instantanés. Pour plus d'informations sur les instantanés et les questions d'espace, reportez-vous à la section [“Comportement d'espace saturé”](#) à la page 62.

La quantité d'espace disque utilisé, disponible et référencé ne comprend pas les modifications en attente. Ces modifications sont généralement prises en compte au bout de quelques secondes. La modification d'un disque utilisant la fonction `fsync(3c)` ou `O_SYNC` ne garantit pas la mise à jour immédiate des informations concernant l'utilisation de l'espace disque.

Les informations de propriété `usedbychildren`, `usedbydataset`, `usedbyreservation` et `usedbysnapshots` peuvent être affichées à l'aide de la commande `zfs list -o space`. Ces propriétés divisent la propriété `used` en espace disque utilisé par les descendants. Pour plus d'informations, reportez-vous au [Tableau 6-1](#).

Propriétés ZFS natives définies

Les propriétés natives définies sont les propriétés dont les valeurs peuvent être récupérées et modifiées. La définition des propriétés natives s'effectue à l'aide de la commande `zfs set`, selon la procédure décrite à la section [“Définition des propriétés ZFS”](#) à la page 206 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 186. 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 [“Définition des quotas et réservations ZFS”](#) à la page 218.

Certaines propriétés natives définies sont spécifiques à un type de jeu de données. Dans ce cas, le type de jeu de données est mentionné dans la description figurant dans le [Tableau 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, reportez-vous à la section “[Propriétés ACL](#)” à la page 247.
- `aclmode`
Pour obtenir une description détaillée, reportez-vous à la section “[Propriétés ACL](#)” à la page 247.
- `atime`
- `canmount`
- Somme de contrôle
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`
Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “[Propriété recordsize](#)” à la page 201.
- `refquota`
- `reservation`
- `reservation`
- `secondarycache`
- `shareiscsi`
- `sharenfs`
- `setuid`
- `snapdir`
- `version`

- `volsize`
Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété `volsize`” à la page 202.
- `volblocksize`
- `zoned`
- `xattr`

Propriété `canmount`

Si la propriété `canmount` est désactivée (valeur `off`), le système de fichiers ne peut pas être monté à l'aide de la commande `zfs mount`, ni de la commande `zfs mount -a`. Définir cette propriété sur `off` est comparable à définir 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 parent à proprement parler n'est jamais monté, ni accessible par les utilisateurs. Dans ce cas, le système de fichiers parent sert de *conteneur* afin de pouvoir définir des propriétés sur le conteneur ; toutefois, le conteneur à proprement parler n'est jamais accessible.

L'exemple suivant illustre la création du système de fichiers `userpool` avec la propriété `canmount` désactivée (valeur `off`). Les points de montage des systèmes de fichiers utilisateur descendants sont définis sur un emplacement commun, `/export/home`. Les systèmes de fichiers descendants héritent des propriétés définies sur le système de fichiers parent, mais celui-ci n'est jamais monté.

```
# zpool create userpool mirror c0t5d0 c1t6d0
# zfs set canmount=off userpool
# zfs set mountpoint=/export/home userpool
# zfs set compression=on userpool
# zfs create userpool/user1
# zfs create userpool/user2
# zfs mount
userpool/user1          /export/home/user1
userpool/user2          /export/home/user2
```

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. Cette valeur de paramètre est utilisée par le logiciel de mise à niveau Oracle Solaris afin que seuls les jeux de données appartenant à l'environnement d'initialisation actif soient montés lors de l'initialisation.

Propriété `recordsize`

La propriété `recordsize` spécifie une taille de bloc suggérée pour les fichiers du système de fichiers.

Cette propriété s'utilise uniquement pour les charges de travail de base de données accédant à des fichiers résidant dans des enregistrements à taille fixe. Le système ZFS ajuste

automatiquement les tailles en fonction d'algorithmes internes optimisés pour les schémas d'accès classiques. Pour les bases de données générant des fichiers volumineux mais accédant uniquement à certains fragments de manière aléatoire, ces algorithmes peuvent se révéler inadaptes. La définition d'une valeur `recordsize` supérieure ou égale à la taille d'enregistrement de la base de données peut améliorer les performances du système de manière significative. Il est vivement déconseillé d'utiliser cette propriété pour les systèmes de fichiers à usage générique. En outre, elle peut affecter les performances du système. La taille spécifiée doit être une puissance de 2 supérieure ou égale à 512 octets et inférieure ou égale à 128 Ko. La modification de la valeur `recordsize` du système de fichiers affecte uniquement les fichiers créés ultérieurement. Cette modification n'affecte pas les fichiers existants.

L'abréviation de la propriété est `recsize`.

Propriété `volsize`

La propriété `volsize` spécifie la taille logique du volume. Par défaut, la création d'un volume définit une réservation de taille identique. Toute modification apportée à la valeur de la propriété `volsize` se répercute dans des proportions identiques au niveau de la réservation. Ce fonctionnement permet d'éviter les comportements inattendus lors de l'utilisation des volumes. L'utilisation de volumes contenant moins d'espace disponible que la valeur indiquée risque, suivant le cas, d'entraîner des comportements non valides et des corruptions de données. Ces symptômes peuvent également survenir lors de la modification et notamment de la réduction de la taille du volume en cours d'utilisation. Faites preuve de prudence lorsque vous ajustez la taille d'un volume.

Même s'il s'agit d'une opération déconseillée, vous avez la possibilité de créer des volumes fragmentés. Pour ce faire, spécifiez l'étiquette `-s` dans la commande `zfs create -V` ou modifiez la réservation, une fois le volume créé. Un *volume fragmenté* désigne un volume dont la réservation est différente de la taille de volume. Les modifications apportées à la propriété `volsize` des volumes fragmentés ne sont pas répercutées au niveau de la réservation.

Pour plus d'informations sur l'utilisation des volumes, reportez-vous à la section "[Volumes ZFS](#)" à la page 281.

Propriétés ZFS définies par l'utilisateur

Outre les propriétés natives, le système ZFS prend en charge des propriétés définies par l'utilisateur. Les propriétés définies par l'utilisateur n'ont aucune incidence sur le comportement du système ZFS. En revanche, elles permettent d'annoter les jeux de données avec des informations adaptées à votre environnement.

Les noms de propriétés définies par l'utilisateur doivent respecter les conventions suivantes :

- Elles doivent contenir le caractère ":" (deux points) afin de les distinguer des propriétés natives.

- Elles doivent contenir des lettres en minuscule, des chiffres ou les signes de ponctuation suivants : ':', '+', ',', '_'.
- La longueur maximale du nom d'une propriété définie par l'utilisateur est de 256 caractères.

La syntaxe attendue des noms de propriétés consiste à regrouper les deux composants suivants (cet espace de noms n'est toutefois pas appliqué par les systèmes ZFS) :

module:property

Si vous utilisez des propriétés définies par l'utilisateur dans un contexte de programmation, spécifiez un nom de domaine DNS inversé pour le composant *module* des noms de propriétés, afin de réduire la probabilité que deux packages développés séparément n'utilisent un nom de propriété identique à des fins différentes. Les noms de propriété commençant par `sun.com` sont réservés à l'usage d'Oracle Corporation.

Les valeurs des propriétés définies par l'utilisateur doivent respecter les conventions suivantes :

- Elles doivent être constituées de chaînes arbitraires systématiquement héritées et elle ne doivent jamais être validées.
- La longueur maximale de la valeur d'une propriété définie par l'utilisateur est de 1024 caractères.

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 valeurs `used`, `available`, `referenced` et `mountpoint` correspondantes. Pour plus d'informations sur ces propriétés, reportez-vous à la section [“Présentation des propriétés ZFS”](#) à la page 189.

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   -
```

Vous pouvez utiliser la commande `zfs list` avec le point de montage d'un système de fichiers.

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 les informations de base sur `tank/home/cha` 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/chua/projects      16K  4.81G  9.0K  /tank/home/chua/projects
tank/home/chua/projects/fs1  8K   4.81G   8K   /tank/home/chua/projects/fs1
tank/home/chua/projects/fs2  8K   4.81G   8K   /tank/home/chua/projects/fs2
```

Pour plus d'informations sur la commande `zfs list`, reportez-vous à la page de manuel [zfs\(1M\)](#).

Création de requêtes ZFS complexes

Les options `o`, `-t` et `-H` permettent de personnaliser la sortie de la commande `zfs list`.

Vous pouvez également personnaliser la sortie des valeurs de propriété en spécifiant l'option `-o` ainsi que la liste des propriétés souhaitées séparées par une virgule. Toute propriété de jeu de données peut être utilisée en tant qu'argument valide. Pour consulter la liste de toutes les propriétés de jeu de données prises en charge, reportez-vous à la section “[Présentation des propriétés ZFS](#)” à la page 189. 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` pour afficher le nom de jeu de données et des valeurs `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/chua     on            /tank/home/chua
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
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. L'option `-H` permet de remplacer les espaces par un caractère de tabulation. Cette option permet notamment d'effectuer des analyses sur les sorties (par exemple, des scripts). L'exemple suivant illustre la sortie de la commande `zfs list` spécifiée avec l'option `-H` :

```
# zfs list -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 206
- “Héritage des propriétés ZFS” à la page 207
- “Envoi de requêtes sur les propriétés ZFS” à la page 208

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 199.

La commande `zfs set` permet d'indiquer une séquence propriété/valeur au format *property=value*, suivie du nom du jeu de données. Lors de chaque appel de la commande `zfs set`, vous ne pouvez définir ou modifier qu'une propriété à la fois.

L'exemple suivant illustre la définition de la propriété `atime` sur la valeur `off` pour `tank/home`.

```
# zfs set atime=off tank/home
```

Vous pouvez également définir les propriétés des systèmes de fichiers une fois ces derniers créés. Exemple :

```
# zfs create -o atime=off tank/home
```

Vous pouvez spécifier les valeurs des propriétés numériques à l'aide de suffixes facile à comprendre (par ordre croissant de grandeur) : BKMGTPEZ. Ces suffixes peuvent être suivis de la lettre b (signifiant "byte", octet) à l'exception du suffixe B, qui fait déjà référence à cette unité de mesure. Les quatre 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 du jeu de données parent (sauf si un quota ou une réservation est explicitement défini pour le jeu de données descendant). Si aucune valeur explicite n'est définie pour une propriété d'un système ascendant, la valeur par défaut de cette propriété est appliquée. Vous pouvez utiliser la commande `zfs inherit` pour effacer la valeur d'une propriété et faire ainsi hériter la valeur du jeu de données 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 inherit`, reportez-vous à la page de manuel [zfs\(1M\)](#).

Envoi de requêtes sur les propriétés ZFS

Le moyen le plus simple pour envoyer une requête sur les valeurs de propriété consiste à exécuter la commande `zfs list`. Pour plus d'informations, reportez-vous à la section [“Affichage des informations de base des systèmes ZFS”](#) à la page 204. Cependant, dans le cadre de requêtes complexes et pour les scripts, utilisez la commande `zfs get` afin de fournir des informations plus détaillées dans un format personnalisé.

La commande `zfs get` permet de récupérer les propriétés de jeu de données. L'exemple suivant illustre la récupération d'une seule valeur de propriété au sein d'un jeu de données :

```
# zfs get checksum tank/ws
NAME          PROPERTY    VALUE      SOURCE
tank/ws      checksum    on         default
```

La quatrième colonne `SOURCE` indique l'origine de la valeur de cette propriété. Le tableau ci-dessous définit les valeurs possibles de la source.

TABEAU 6-3 Valeurs possibles de la colonne `SOURCE` (commande `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 jeu de données parent spécifié par la chaîne <i>dataset-name</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> .

TABLEAU 6-3 Valeurs possibles de la colonne SOURCE (commande `zfs get`) (Suite)

Valeur	Description
temporary	Cette 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 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 215.
-(none)	Cette propriété est en lecture seule. Sa valeur est générée par ZFS.

Le mot-clé `all` permet de récupérer toutes les valeurs de propriétés du jeu de données. Les exemples suivants utilisent le mot-clé `all` :

```
# zfs get all tank/home
NAME      PROPERTY      VALUE      SOURCE
tank/home type          filesystem -
tank/home creation    Tue Jun 29 11:44 2010 -
tank/home used        21K        -
tank/home available 66.9G     -
tank/home referenced 21K        -
tank/home compressratio 1.00x     -
tank/home mounted    yes        -
tank/home quota      none       default
tank/home reservation none       default
tank/home recordsize 128K      default
tank/home mountpoint /tank/home default
tank/home sharenfs   off        default
tank/home checksum   on         default
tank/home compression off        default
tank/home atime      on         default
tank/home devices   on         default
tank/home exec       on         default
tank/home setuid     on         default
tank/home readonly  off        default
tank/home zoned      off        default
tank/home snapdir   hidden    default
tank/home aclmode   groupmask default
tank/home aclinherit restricted  default
tank/home canmount  on         default
tank/home shareiscsi off        default
tank/home xattr     on         default
tank/home copies    1          default
tank/home version   4          -
tank/home utf8only  off        -
tank/home normalization none       -
tank/home casesensitivity sensitive -
tank/home vscan     off        default
tank/home nbmand    off        default
tank/home sharesmb  off        default
tank/home refquota  none       default
tank/home refreservation none      default
tank/home primarycache all        default
tank/home secondarycache all        default
tank/home usedbysnapshots 0          -
tank/home usedbydataset 21K        -
```

tank/home	usedbychildren	0	-
tank/home	usedbyreservation	0	-
tank/home	logbias	latency	default

Remarque – Le fonctionnement des propriétés `casesensitivity`, `nbmand`, `normalization`, `sharesmb`, `utf8only` et `vscan` n'est pas optimal dans la version Solaris 10 car le service Oracle Solaris SMB n'est pas pris en charge dans la version Solaris 10.

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
```

Vous pouvez interroger les valeurs d'une propriété à l'aide de la commande `zfs get` sans spécifier le système de fichiers cible (la commande fonctionne sur tous les systèmes de fichiers et les pools). 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. Vous pouvez utiliser l'option `-H` pour omettre les informations d'en-tête et pour remplacer un espace par un caractère de tabulation. L'uniformisation des espaces permet de faciliter l'analyse des données. Vous pouvez utiliser l'option `-o` pour personnaliser la sortie de l'une des façons suivantes :

- Le littéral nom peut être utilisé avec une liste séparée par des virgules de propriétés comme l'explique la section “[Présentation des propriétés ZFS](#)” à la page 189.
- Une liste de champs littéraux séparés par des virgules (name, value, property et source) suivi d'un espace et d'un argument. En d'autres termes, il s'agit d'une liste de propriétés séparées par des virgules.

L'exemple suivant illustre la commande permettant de récupérer une seule valeur en spécifiant les options -H et -o de la commande `zfs get`:

```
# zfs get -H -o value compression tank/home
on
```

L'option -p indique les valeurs numériques sous leur forme exacte. Par exemple, 1 Mo serait signalé sous la forme 1000000. Cette option peut être utilisée comme suit :

```
# zfs get -H -o value -p used tank/home
182983742
```

L'option -r permet de récupérer de manière récursive les valeurs demandées pour l'ensemble des descendants et peut s'utiliser avec toutes les options mentionnées précédemment. Dans l'exemple suivant, les options -H, -o et -r sont spécifiées afin de récupérer le nom du 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 211
- “[Montage de système de fichiers ZFS](#)” à la page 214
- “[Utilisation de propriétés de montage temporaires](#)” à la page 215
- “[Démontage des systèmes de fichiers ZFS](#)” à la page 215
- “[Activation et annulation du partage des systèmes de fichiers ZFS](#)” à la page 216

Gestion des points de montage ZFS

Par défaut, un système de fichiers ZFS est automatiquement monté lors de sa création. Vous pouvez déterminer un comportement de point de montage spécifique pour un système de fichiers comme décrit dans cette section.

Vous pouvez également définir le point de montage par défaut du jeu de données du pool lors de l'exécution de la commande de création `zpool create` en spécifiant l'option `-m`. Pour plus d'informations sur la création de pools, reportez-vous à la section [“Création d'un pool de stockage ZFS”](#) à la page 72.

Tous les systèmes de fichiers ZFS sont montés lors de l'initialisation à l'aide du service `svc://system/filesystem/local SMF` (Service Management Facility). Les systèmes de fichiers sont montés sous `/path`, où `path` correspond au nom du système de fichiers. Système de fichiers ZFS

Vous pouvez remplacer le point de montage par défaut à l'aide de la commande `zfs set` pour définir la propriété `mountpoint` sur un chemin spécifique. Le système de fichiers ZFS crée automatiquement le point de montage spécifié, le cas échéant. Il monte ensuite automatiquement le système de fichiers correspondant, lors de l'exécution de la commande `zfs mount -a`. Il n'est pas nécessaire de modifier le fichier `/etc/vfstab`.

La propriété `mountpoint` est héritée. Par exemple, si le fichier `pool/home` est doté d'une propriété `mountpoint` définie sur `/export/stuff`, alors `pool/home/user` hérite de la valeur `/export/stuff/user` pour sa propriété `mountpoint`.

Pour éviter le montage d'un système de fichiers, définissez la propriété `mountpoint` sur `none`. En outre, la propriété `canmount` peut être utilisée pour contrôler le montage d'un système de fichiers. Pour plus d'informations sur la propriété `canmount`, reportez-vous à la section [“Propriété canmount”](#) à la page 201.

Les systèmes de fichiers peuvent également être gérés de manière explicite à l'aide d'interfaces de montage héritées en utilisant la commande `zfs set` pour définir la propriété `mountpoint` sur `legacy`. Dans ce cas, le montage et la gestion d'un système de fichiers ne sont pas gérés automatiquement par ZFS. Ces opérations s'effectuent alors à l'aide des outils hérités, comme les commandes `mount` et `umount` et le fichier `/etc/vfstab`. Pour plus d'informations sur les montages hérités, reportez-vous à la section [“Points de montage hérités”](#) à la page 213.

Points de montage automatiques

- Lorsque vous modifiez la propriété `mountpoint` de `legacy` à `none` sur un chemin spécifique, le système de fichiers ZFS est automatiquement monté.
- Si le système de fichiers ZFS est géré automatiquement sans être monté et si la propriété `mountpoint` est modifiée, le système de fichiers reste démonté.

Les 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. Les répertoires de point de montage sont créés, le cas échéant. Si ZFS n'est pas en mesure de démonter un système de fichiers parce qu'il est actif, une erreur est signalée et un démontage manuel forcé doit être effectué.

Points de montage hérités

La gestion des systèmes de fichiers ZFS peut s'effectuer à l'aide d'outils hérités. Pour cela, la propriété `mountpoint` doit être définie sur `legacy`. Les systèmes de fichiers hérités sont alors gérés à l'aide des commandes `mount` et `umount` et du fichier `/etc/vfstab`. Lors de l'initialisation, le système de fichiers ZFS ne monte pas automatiquement les systèmes de fichiers hérités et les commandes ZFS `mount` et `umount` ne fonctionnent pas sur ces types de jeu 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
```

Pour monter automatiquement un système de fichiers hérité lors de l'initialisation, vous devez ajouter une entrée au fichier `/etc/vfstab`. L'exemple suivant montre l'entrée telle qu'elle peut apparaître dans le fichier `/etc/vfstab` :

```
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot  options
#
tank/home/eschrock -          /mnt       zfs     -        yes      -
```

Les entrées `device to fsck` et `fsck pass` sont définies sur `-` car la commande `fsck` ne s'applique pas aux systèmes de fichiers ZFS. Pour plus d'informations sur l'intégrité des données ZFS, reportez-vous à la section [“Sémantique transactionnelle” à la page 47](#).

Montage de système de fichiers ZFS

Le montage des systèmes de fichiers ZFS s'effectue automatiquement lors du processus de création ou lors de l'initialisation du système. Vous ne devez utiliser la commande `zfs mount` que lorsque vous devez modifier les options de montage ou monter/démonter explicitement les systèmes de fichiers.

Spécifiée sans argument, la commande `zfs mount` répertorie tous les systèmes de fichiers actuellement montés gérés par ZFS. Les points de montage hérités ne sont pas inclus. 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 -O flag
# zfs mount -O 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 :

TABLEAU 6-4 Options et propriétés de montage ZFS

Propriétés	Option de montage
<code>atime</code>	<code>atime/noatime</code>
<code>devices</code>	<code>devices/nodevices</code>
<code>exec</code>	<code>exec/noexec</code>

TABLEAU 6-4 Options et propriétés de montage ZFS (Suite)

Propriétés	Option de montage
nbmand	nbmand/nonbmand
readonly	ro/rw
setuid	setuid/nosetuid
xattr	xattr/noaxttr

L'option de montage nosuid représente un alias de nodevices, nosetuid.

Utilisation de propriétés de montage temporaires

Si les options de montage décrites à la section précédente sont définies de manière explicite en spécifiant l'option `-o` avec la commande `zfs mount`, les valeurs des propriétés associées sont remplacées de manière temporaire. Ces valeurs de propriété sont désignées par la chaîne `temporary` dans la commande `zfs get` et reprennent leur valeur d'origine une fois le système de fichiers démonté. Si une valeur de propriété est modifiée alors que le 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`. Le système de fichiers est censé être démonté.

```
# zfs mount -o ro tank/home/perrin
```

Pour modifier temporairement une valeur de propriété sur un système de fichiers monté, vous devez utiliser l'option spécifique `remount`. Dans l'exemple suivant, la propriété `atime` est temporairement définie sur la valeur `off` pour un système de fichiers monté :

```
# zfs mount -o remount,noatime 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 ZFS peut s'effectuer à l'aide de la commande `zfs unmount`. La commande `unmount` peut utiliser le point de montage ou le nom du système de fichiers comme argument.

L'exemple suivant illustre le démontage d'un système de fichiers avec l'argument de nom de système de fichiers :

```
# zfs unmount 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 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

Le système de fichiers ZFS permet de partager automatiquement des systèmes de fichiers en définissant la propriété `sharenfs`. Cette propriété évite d'avoir à modifier le fichier `/etc/dfs/dfstab` lors de l'ajout d'un système de fichiers. La propriété `sharenfs` correspond à une liste d'options séparées par une virgule spécifiée avec la commande `share`. La valeur `on` constitue un alias des options de partage par défaut, qui attribuent les 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 `sharenfs` est héritée et les systèmes de fichiers sont automatiquement partagés lorsqu'ils sont créés (si la propriété héritée n'est pas définie sur `off`). Exemple :


```
# zfs set sharenfs=on tank/home
# zfs create tank/home/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

Même si la plupart des systèmes de fichiers sont automatiquement partagés ou non lors de l'initialisation, la création ou la destruction de ces derniers, il est parfois nécessaire de les définir explicitement comme non partagés. Ce type d'opération s'effectue à l'aide de la commande `zfs unshare`. 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

La plupart du temps, le comportement automatique du système de fichiers ZFS par rapport au partage de système de fichiers lors de l'initialisation et de la création de ces derniers est suffisant pour les opérations normales. Si vous êtes amené à annuler le partage d'un système de fichiers, vous pouvez réactiver le partage à l'aide de la commande `zfs share`. 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 `mount` héritée, les commandes `share` et `unshare` peuvent être exécutées sur les systèmes de fichiers ZFS. Vous pouvez dès lors partager manuellement un système de fichiers en spécifiant des options différentes de celles définies pour la propriété

sharenfs. Ce modèle de gestion est déconseillé. La gestion des partages NFS doit s'effectuer intégralement à l'aide du système ZFS ou intégralement à l'aide du fichier `/etc/dfs/dfstab`. Le modèle ZFS a été conçu pour simplifier et pour faciliter les opérations de gestion par rapport au modèle classique.

Définition des quotas et réservations ZFS

La propriété `quota` permet de limiter la quantité d'espace disque disponible pour un système de fichiers. La propriété `reservation` permet quant à elle de garantir la disponibilité d'une certaine quantité d'espace disque pour un système de fichiers. Ces deux propriétés s'appliquent au 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 disque totale utilisée par `tank/home` et par tous ses descendants ne peut pas excéder le `quota` défini. De même, si une `reservation` est définie pour le jeu de données `tank/home`, cette `reservation` s'applique à `tank/home` et à tous ses descendants. La quantité d'espace disque utilisée par un jeu de données et par tous ses descendants est indiquée par la propriété `used`.

Les propriétés `refquota` et `refreservation` vous permettent de gérer l'espace d'un système de fichiers sans prendre en compte l'espace disque utilisé par les descendants, notamment les instantanés et les clones.

Dans cette version de Solaris, vous pouvez définir un `quota` d'utilisateur (*user*) ou de *groupe* sur la quantité d'espace disque utilisée par les fichiers appartenant à un utilisateur ou à un groupe spécifique. Les propriétés de `quota` d'utilisateur et de *groupe* ne peuvent pas être définies sur un volume, sur un système de fichiers antérieur à la version 4, ou sur un *pool* antérieur à la version 15.

Considérez les points suivants pour déterminer quelles fonctions de `quota` et de `reservation` conviennent le mieux à la gestion de vos systèmes de fichiers :

- Les propriétés `quota` et `reservation` conviennent à la gestion de l'espace disque utilisé par les jeux de données et leurs descendants.
- Les propriétés `refquota` et `refreservation` conviennent à la gestion de l'espace disque utilisé par les jeux de données.
- La définition d'une propriété `refquota` ou `refreservation` supérieure à une la propriété `quota` ou `reservation` n'a aucun effet. Lorsque vous définissez la propriété `quota` ou `refquota`, les opérations qui tentent de dépasser l'une de ces valeurs échouent. Il est possible de dépasser une valeur `quota` supérieure à une valeur `refquota`. Par exemple, si certains blocs d'instantanés sont modifiés, la valeur `quota` risque d'être dépassée avant la valeur `refquota`.
- Les `quotas` d'utilisateurs et de groupes permettent d'augmenter plus facilement l'espace disque contenant de nombreux comptes d'utilisateur, par exemple dans une université.

Pour plus d'informations sur la définition de quotas et réservations, reportez-vous aux sections “Définitions de quotas sur les systèmes de fichiers ZFS” à la page 219 et “Définition de réservations sur les systèmes de fichiers ZFS” à la page 222.

Définitions de quotas sur les systèmes de fichiers ZFS

Les quotas des systèmes de fichiers ZFS peuvent être définis et affichés à l'aide des commandes `zfs set` et `zfs get`. Dans l'exemple suivant, un quota de 10 Go est défini pour `tank/home/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 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 disque disponible. Toutefois, `tank/home/bonwick` et `tank/home/bonwick/ws` disposent uniquement de 10 Go d'espace disponible, respectivement, 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 propriété `refquota` sur un jeu de données pour limiter l'espace disque occupé par le jeu de données. Cette limite fixe ne comprend pas l'espace disque utilisé par les descendants. 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 disque 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 studentA peut supprimer des fichiers pour libérer de l'espace même en présence d'instantanés.

Dans l'exemple précédent, le plus petit des deux quotas (10 Go par rapport à 20 Go) s'affiche dans la sortie `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. Exemple :

```

# zfs get refquota,quota students/studentA
NAME                PROPERTY VALUE          SOURCE
students/studentA  refquota 10G            local
students/studentA  quota    20G            local

```

Définition de quotas d'utilisateurs et de groupes sur un système de fichiers ZFS

Vous pouvez définir un quota d'utilisateurs ou de groupes en utilisant respectivement les commandes `zfs userquota` et `zfs groupquota`. Exemple :

```

# zfs create students/compsci
# zfs set userquota@student1=10G students/compsci
# zfs create students/labstaff
# zfs set groupquota@staff=20GB students/labstaff

```

Affichez le quota d'utilisateurs ou de groupes actuel comme suit :

```

# zfs get userquota@student1 students/compsci
NAME                PROPERTY VALUE          SOURCE
students/compsci  userquota@student1 10G            local
# zfs get groupquota@staff students/labstaff
NAME                PROPERTY VALUE          SOURCE
students/labstaff  groupquota@staff 20G            local

```

Vous pouvez afficher l'utilisation générale de l'espace disque par les utilisateurs et les groupes en interrogeant les propriétés suivantes :

```
# zfs userspace students/compsci
TYPE      NAME      USED  QUOTA
POSIX User  root      227M  none
POSIX User  student1  455M  10G
# zfs groupspace students/labstaff
TYPE      NAME      USED  QUOTA
POSIX Group root      217M  none
POSIX Group staff   217M  20G
```

Pour identifier l'utilisation de l'espace disque d'un groupe ou d'un utilisateur, vous devez interroger les propriétés suivantes :

```
# zfs get userused@student1 students/compsci
NAME                PROPERTY          VALUE              SOURCE
students/compsci   userused@student1 455M              local
# zfs get groupused@staff students/labstaff
NAME                PROPERTY          VALUE              SOURCE
students/labstaff  groupused@staff   217M              local
```

Les propriétés de quota d'utilisateurs et de groupes ne sont pas affichées à l'aide de la commande `zfs get all dataset`, qui affiche une liste de toutes les autres propriétés du système de fichiers.

Vous pouvez supprimer un quota d'utilisateurs ou de groupes comme suit :

```
# zfs set userquota@user1=none students/compsci
# zfs set groupquota@staff=none students/labstaff
```

Les quotas d'utilisateurs et de groupes sur les systèmes de fichiers ZFS offrent les fonctionnalités suivantes :

- Un quota d'utilisateurs ou de groupes défini sur un système de fichiers parent n'est pas automatiquement hérité par un système de fichiers descendant.
- Cependant, le quota d'utilisateurs ou de groupes est appliqué lorsqu'un clone ou un instantané est créé à partir d'un système de fichiers lié à un quota d'utilisateurs ou de groupes. De même, un quota d'utilisateurs ou de groupes est inclus avec le système de fichiers lorsqu'un flux est créé à l'aide de la commande `zfs send`, même sans l'option `-R`.
- Les utilisateurs dénués de privilèges peuvent uniquement disposer de leur propre utilisation d'espace disque. L'utilisateur `root` ou l'utilisateur qui s'est vu accorder le privilège `userused` ou `groupused` peut accéder aux informations de comptabilité de l'espace disque utilisateur ou groupe de tout le monde.
- Les propriétés `userquota` et `groupquota` ne peuvent pas être définies sur les volumes ZFS, sur un système de fichiers antérieur à la version 4, ou sur un pool antérieur à la version 15.

L'application des quotas d'utilisateurs et de groupes peut être différée de quelques secondes. Ce délai signifie que les utilisateurs peuvent dépasser leurs quotas avant que le système ne le remarque et refuse d'autres écritures en affichant le message d'erreur `EDQUOT` .

Vous pouvez utiliser la commande `quota` héritée pour examiner les quotas d'utilisateurs dans un environnement NFS où un système de fichiers ZFS est monté, par exemple. Sans aucune option, la commande `quota` affiche uniquement la sortie en cas de dépassement du quota de l'utilisateur. Exemple :

```
# zfs set userquota@student1=10m students/compsci
# zfs userspace students/compsci
TYPE      NAME      USED  QUOTA
POSIX User root      227M  none
POSIX User student1 455M  10M
# quota student1
Block limit reached on /students/compsci
```

Si vous réinitialisez le quota d'utilisateurs et que la limite du quota n'est plus dépassée, vous devez utiliser la commande `quota -v` pour examiner le quota de l'utilisateur. Exemple :

```
# zfs set userquota@student1=10GB students/compsci
# zfs userspace students/compsci
TYPE      NAME      USED  QUOTA
POSIX User root      227M  none
POSIX User student1 455M  10G
# quota student1
# quota -v student1
Disk quotas for student1 (uid 201):
Filesystem      usage quota limit      timeleft files quota limit      timeleft
/students/compsci
                466029 10485760 10485760
```

Définition de réservations sur les systèmes de fichiers ZFS

Une *réserve* ZFS désigne une quantité d'espace disque du pool garantie pour un jeu de données. Dès lors, pour réserver une quantité d'espace disque pour un jeu de données, cette quantité doit être actuellement disponible sur le pool. La quantité totale d'espace non utilisé des réservations ne peut pas dépasser la quantité d'espace disque non utilisé du pool. La définition et l'affichage des réservations ZFS s'effectuent respectivement à l'aide des commandes `zfs set` et `zfs get`. Exemple :

```
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME          PROPERTY  VALUE  SOURCE
tank/home/moore reservation 5G     local
```

Les réservations 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 33.5G 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 le calcul de l'espace disque utilisé des jeux de données parent et non dans le quota, la réservation ou les deux.

```
# zfs set quota=5G pool/filesystem
# zfs set reservation=10G pool/filesystem/user1
cannot set reservation for 'pool/filesystem/user1': size is greater than
available space
```

Un jeu de données peut utiliser davantage d'espace disque que sa réservation, du moment que le pool dispose d'un espace non réservé et disponible et que l'utilisation actuelle du jeu de données se trouve en dessous des quotas. Un jeu de données ne peut pas utiliser un espace disque réservé à un autre jeu de données.

Les réservations ne sont pas cumulatives. En d'autres termes, l'exécution d'une nouvelle commande `zfs set` pour un jeu de données déjà associé à une réservation n'entraîne pas l'ajout de la nouvelle réservation à la réservation existante. La seconde réservation remplace la première. Exemple :

```
# 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 réservation `refreservation` pour garantir un espace disque ne contenant aucun instantané ou clone au jeu de données. Cette valeur est prise en compte dans le calcul de l'espace utilisé des jeux de données parent et vient en déduction des quotas et réservations des jeux de données parent. 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 le calcul de l'espace utilisé du parent.

Dans l'exemple précédent, le plus petit des deux quotas (10 Go par rapport à 20 Go) s'affiche dans la sortie `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. Exemple :

```
# zfs get reservation,refreserv profs/prof1
NAME          PROPERTY      VALUE      SOURCE
profs/prof1   reservation   20G       local
profs/prof1   refreserv     10G       local
```

Lorsque la propriété `refreservation` est définie, un instantané n'est autorisé que si suffisamment d'espace non réservé est disponible dans le pool au-delà de cette réservation afin de pouvoir contenir le nombre actuel d'octets *référéncés* dans le jeu de données.

Utilisation des instantanés et des clones ZFS Oracle Solaris

Ce chapitre fournit des informations sur la création et la gestion d'instantanés et de clones ZFS Oracle Solaris. Des informations concernant l'enregistrement des instantanés sont également fournies.

Il contient les sections suivantes :

- “Présentation des instantanés ZFS” à la page 225
- “Création et destruction d'instantanés ZFS” à la page 226
- “Affichage et accès des instantanés ZFS” à la page 229
- “Restauration d'un instantané ZFS” à la page 231
- “Présentation des clones ZFS” à la page 232
- “Création d'un clone ZFS” à la page 232
- “Destruction d'un clone ZFS” à la page 233
- “Remplacement d'un système de fichiers ZFS par un clone ZFS” à la page 233
- “Envoi et réception de données ZFS” à la page 234

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 disque.

Les instantanés ZFS présentent les caractéristiques suivantes :

- Persistance au cours des réinitialisations de système.
- Théoriquement, le nombre maximal d'instantanés est de 2^{64} instantanés.
- Les instantanés n'utilisent aucune sauvegarde de secours distincte. Les instantanés consomment de l'espace disque provenant directement du pool de stockage auquel appartient le système de fichiers ou le volume à partir duquel ils ont été créés.

- Une seule opération, dite atomique, permet de créer rapidement des instantanés récursifs. Ceux-ci sont tous créés simultanément ou ne sont pas créés du tout. Grâce à ce type d'opération d'instantané atomique, une certaine cohérence des données d'instantané est assurée, y compris pour les systèmes de fichiers descendants.

Il n'est pas possible d'accéder directement aux instantanés de volumes, mais ils peuvent être clonés, sauvegardés, restaurés, etc. Pour plus d'informations sur la sauvegarde d'un instantané ZFS, reportez-vous à la section [“Envoi et réception de données ZFS”](#) à la page 234.

- [“Création et destruction d'instantanés ZFS”](#) à la page 226
- [“Affichage et accès des instantanés ZFS”](#) à la page 229
- [“Restauration d'un instantané ZFS”](#) à la page 231

Création et destruction d'instantanés ZFS

La commande `zfs snapshot` permet de créer les instantanés. Elle ne prend pour argument que le nom de l'instantané à créer. Le nom de l'instantané est spécifié comme suit :

```
filesystem@snapname
volume@snapname
```

Ce nom doit respecter les conventions d'attribution de nom définies à la section [“Exigences d'attribution de noms de composants ZFS”](#) à la page 52.

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
rpool/ROOT/zfs2BE@zfs2BE           78.3M  -      4.53G  -
tank/home@now                       0      -      26K    -
tank/home/ahrens@now                0      -      259M   -
tank/home/anne@now                  0      -      156M   -
tank/home/bob@now                    0      -      156M   -
tank/home/cindys@now                 0      -      104M   -
```

Les propriétés des instantanés ne sont pas modifiables. Les propriétés des jeux de données ne peuvent pas être appliquées à un instantané. Exemple :

```
# zfs set compression=on tank/home/ahrens@now
cannot set compression property for 'tank/home/ahrens@now': snapshot
properties cannot be modified
```

La commande `zfs destroy` permet de détruire les instantanés. Exemple :

```
# zfs destroy tank/home/ahrens@now
```

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 187.

Conservation des clichés ZFS

Si vous disposez de plusieurs stratégies automatiques pour les instantanés pour que l'instantané le plus ancien soit par exemple détruit par la commande `zfs receive` car il n'existe plus du côté de l'envoi, vous pouvez utiliser la fonction de conservation des instantanés.

La *conservation* d'un instantané empêche sa destruction. En outre, cette fonction permet de supprimer un instantané contenant des clones en attendant la suppression du dernier clone à l'aide de la commande `zfs destroy -d`. Chaque instantané est associé à un décompte de référence utilisateur initialisé sur 0 (zéro). Ce nombre augmente de 1 à chaque fois qu'un instantané est conservé et diminue de 1 à chaque fois qu'un instantané conservé est libéré.

Dans la version précédente de Solaris, les instantanés ne peuvent pas être détruits à l'aide de la commande `zfs destroy` s'ils ne contiennent aucun clone. Dans cette version de Solaris, les instantanés doivent également renvoyer un décompte de référence utilisateur égal à 0 (zéro).

Vous pouvez conserver un instantané ou un jeu d'instantanés. Par exemple, la syntaxe suivante insère une balise de conservation `keep` sur `citerne/home/cindys/snap@1`.

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

Vous pouvez utiliser l'option `-r` pour conserver récursivement les instantanés de tous les systèmes de fichiers descendants. Exemple :

```
# zfs snapshot -r tank/home@now
# zfs hold -r keep tank/home@now
```

Cette syntaxe permet d'ajouter une référence `keep` unique à cet instantané ou à ce jeu d'instantanés. Chaque instantané possède son propre espace de noms de balise dans lequel chaque balise de conservation doit être unique. Si un instantané est conservé, les tentatives de destruction de ce dernier à l'aide de la commande `zfs destroy` échoueront. Exemple :

```
# zfs destroy tank/home/cindys@snap1
cannot destroy 'tank/home/cindys@snap1': dataset is busy
```

Si vous souhaitez détruire un instantané conservé, utilisez l'option `-d`. Exemple :

```
# zfs destroy -d tank/home/cindys@snap1
```

Utilisez la commande `zfs holds` pour afficher la liste des instantanés conservés. Exemple :

```
# zfs holds tank/home@now
NAME          TAG      TIMESTAMP
tank/home@now keep    Thu Jul 15 11:25:39 2010
```

```
# zfs holds -r tank/home@now
NAME          TAG      TIMESTAMP
tank/home/cindys@now keep    Thu Jul 15 11:25:39 2010
tank/home/mark@now keep    Thu Jul 15 11:25:39 2010
tank/home@now keep    Thu Jul 15 11:25:39 2010
```

Vous pouvez utiliser la commande `zfs release` pour libérer un instantané ou un jeu d'instantanés conservé. Exemple :

```
# zfs release -r keep tank/home@now
```

Si l'instantané est libéré, l'instantané peut être détruit à l'aide de la commande `zfs destroy`. Exemple :

```
# zfs destroy -r tank/home@now
```

Deux nouvelles propriétés permettent d'identifier les informations de conservation d'un instantané :

- La propriété `defer_destroy` est définie sur `on` si l'instantané a été marqué en vue d'une destruction différée à l'aide de la commande `zfs destroy -d`. Dans le cas contraire, la propriété est définie sur `off`.
- La propriété `userrefs` est définie sur le nombre de conservations d'instantanés, également appelée décompte de références utilisateur.

Renommage d'instantanés ZFS

Vous pouvez renommer les instantanés. Cependant, ils doivent rester dans le même pool et dans le même jeu de données dans lequel il ont été créés. Exemple :

```
# zfs rename tank/home/cindys@083006 tank/home/cindys@today
```

En outre, la syntaxe raccourci suivante est équivalente à la syntaxe précédente :

```
# zfs rename tank/home/cindys@083006 today
```

L'opération de renommage (rename) d'instantané n'est pas prise en charge, car le nom du pool cible et celui du système de fichiers ne correspondent pas au pool et au système de fichiers dans lesquels l'instantané a été créé :

```
# zfs rename tank/home/cindys@today pool/home/cindys@aturday
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  MOUNTPOINT
users                               270K  16.5G  22K    /users
users/home                          76K   16.5G  22K    /users/home
users/home@yesterday                0     -     22K    -
users/home/markm                   18K   16.5G  18K    /users/home/markm
users/home/markm@yesterday          0     -     18K    -
users/home/marks                   18K   16.5G  18K    /users/home/marks
users/home/marks@yesterday          0     -     18K    -
users/home/neil                    18K   16.5G  18K    /users/home/neil
users/home/neil@yesterday           0     -     18K    -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -r users/home
NAME                                USED  AVAIL  REFER  MOUNTPOINT
users/home                          76K   16.5G  22K    /users/home
users/home@2daysago                0     -     22K    -
users/home/markm                   18K   16.5G  18K    /users/home/markm
users/home/markm@2daysago          0     -     18K    -
users/home/marks                   18K   16.5G  18K    /users/home/marks
users/home/marks@2daysago          0     -     18K    -
users/home/neil                    18K   16.5G  18K    /users/home/neil
users/home/neil@2daysago           0     -     18K    -
```

Affichage et accès des instantanés ZFS

Vous pouvez activer ou désactiver l'affichage des listes d'instantanés de la sortie `zfs list` en utilisant la propriété de `pool listsnapshots`. Cette propriété est activée par défaut.

Si vous désactivez cette propriété, vous pouvez utiliser la commande `zfs list -t snapshot` pour afficher les informations relatives à un instantané. Ou activez la propriété de `pool listsnapshots`. Exemple :

```
# zpool get listsnapshots tank
NAME  PROPERTY  VALUE  SOURCE
tank  listsnapshots  on     default
# zpool set listsnapshots=off tank
# zpool get listsnapshots tank
NAME  PROPERTY  VALUE  SOURCE
tank  listsnapshots  off     local
```

Les instantanés des systèmes de fichiers sont accessibles dans le répertoire `.zfs/snapshot` au sein de la racine du système de fichiers. 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 Jun 30 16:16 2010
tank/home/ahrens@now                Wed Jun 30 16:16 2010
tank/home/anne@now                  Wed Jun 30 16:16 2010
tank/home/bob@now                    Wed Jun 30 16:16 2010
tank/home/cindys@now                 Wed Jun 30 16:16 2010
```

Comptabilisation de l'espace disque des instantanés ZFS

Lors de la création d'un instantané, son espace disque est initialement partagé entre l'instantané et le système de fichiers et éventuellement avec des instantanés précédents. Lorsque le système de fichiers change, l'espace disque précédemment partagé devient dédié à l'instantané, et il est compté dans la propriété `used` de l'instantané. De plus, la suppression d'instantanés peut augmenter la quantité d'espace disque dédié à d'autres instantanés (et, par conséquent, *utilisé* par ceux-ci).

La valeur de la propriété `referenced` de l'espace d'un instantané est la même que lors de la création de l'instantané sur le système de fichiers.

Vous pouvez identifier des informations supplémentaires sur la façon dont les valeurs de la propriété `used` sont utilisées. Les nouvelles propriétés de système de fichiers en lecture seule décrivent l'utilisation de l'espace disque pour les clones, les systèmes de fichiers et les volumes. Exemple :

```
$ zfs list -o space
# zfs list -ro space tank/home
NAME                                AVAIL  USED  USED SNAP  USED DS  USED REFRESERV  USED CHILD
```

tank/home	66.3G	675M	0	26K	0	675M
tank/home@now	-	0	-	-	-	-
tank/home/ahrens	66.3G	259M	0	259M	0	0
tank/home/ahrens@now	-	0	-	-	-	-
tank/home/anne	66.3G	156M	0	156M	0	0
tank/home/anne@now	-	0	-	-	-	-
tank/home/bob	66.3G	156M	0	156M	0	0
tank/home/bob@now	-	0	-	-	-	-
tank/home/cindys	66.3G	104M	0	104M	0	0
tank/home/cindys@now	-	0	-	-	-	-

Pour une description de ces propriétés, reportez-vous au [Tableau 6-1](#).

Restauration d'un instantané ZFS

Vous pouvez utiliser la commande `zfs rollback` pour abandonner toutes les modifications apportées à un système de fichiers depuis la création d'un instantané spécifique. Le système de fichiers revient à l'état dans lequel il était lors de la prise de l'instantané. Par défaut, la commande ne permet pas de restaurer un instantané autre que le plus récent.

Pour restaurer un instantané précédent, tous les instantanés intermédiaires doivent être détruits. Vous pouvez détruire les instantanés précédents en spécifiant l'option `-r`.

S'il existe des clones d'un instantané intermédiaire, vous devez spécifier l'option `-R` pour détruire également les clones.

Remarque – Si le système de fichiers que vous souhaitez restaurer est actuellement monté, il doit être démonté, puis remonté. Si le système de fichiers ne peut pas être démonté, la restauration échoue. L'option `-f` force le démontage du système de fichiers, le cas échéant.

Dans l'exemple suivant, 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 cet exemple, les instantanés `wednesday` et `thursday` sont détruits en raison de la restauration de l'instantané `tuesday` précédent.

```
# zfs list -r -t snapshot -o name,creation tank/home/ahrens
NAME                CREATION
tank/home/ahrens@now Wed Jun 30 16:16 2010
```

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.

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 206.

Dans la mesure où un clone partage initialement son espace disque avec l'instantané d'origine, la valeur de la propriété `used` est initialement égale à zéro. À mesure que le clone est modifié, il utilise de plus en plus d'espace disque. La propriété `used` de l'instantané d'origine ne tient pas compte de l'espace disque consommé par le clone.

- “[Création d'un clone ZFS](#)” à la page 232
- “[Destruction d'un clone ZFS](#)” à la page 233
- “[Remplacement d'un système de fichiers ZFS par un clone ZFS](#)” à la page 233

Création d'un clone ZFS

Pour créer un clone, utilisez la commande `zfs clone` en spécifiant l'instantané à partir duquel créer le clone, ainsi que le nom du nouveau volume ou système de fichiers. Le nouveau volume ou système de fichiers peut se trouver à tout emplacement de la hiérarchie ZFS. Le nouveau jeu de données est du même type (un système de fichiers ou un volume, par exemple) que celui de l'instantané à partir duquel le clone a été créé. Vous ne pouvez pas créer le clone d'un système de fichiers dans un autre pool que celui de l'instantané du système de fichiers d'origine.

Dans l'exemple suivant, un nouveau clone appelé `tank/home/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 : Destruction

```
# 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 *original* devienne le clone du système de fichiers spécifié. En outre, cette fonction permet de détruire le système de fichiers à partir duquel le clone a été créé. Il est impossible de détruire un système de fichiers d'origine possédant des clones actifs, sans le remplacer par l'un de ses clones. Pour plus d'informations sur la destruction des clones, reportez-vous à la section “[Destruction d'un clone ZFS](#)” à la page 233.

Dans l'exemple suivant, le système de fichiers `tank/test/productA` est cloné, puis le clone du système de fichiers (`tank/test/productAbeta`) devient le système de fichiers `tank/test/productA` d'origine.

```
# zfs create tank/test
# zfs create tank/test/productA
# zfs snapshot tank/test/productA@today
# zfs clone tank/test/productA@today tank/test/productAbeta
# zfs list -r tank/test
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank/test	104M	66.2G	23K	/tank/test
tank/test/productA	104M	66.2G	104M	/tank/test/productA
tank/test/productA@today	0	-	104M	-
tank/test/productAbeta	0	66.2G	104M	/tank/test/productAbeta

```
# zfs promote tank/test/productAbeta
# zfs list -r tank/test
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank/test	104M	66.2G	24K	/tank/test
tank/test/productA	0	66.2G	104M	/tank/test/productA
tank/test/productAbeta	104M	66.2G	104M	/tank/test/productAbeta
tank/test/productAbeta@today	0	-	104M	-

Dans la sortie `zfs list`, les informations de comptabilisation de l'espace disque du système de fichiers d'origine `productA` ont été remplacées par celles du système de fichiers `productAbeta`

Pour terminer le processus de remplacement de clone, renommez les systèmes de fichiers.
Exemple :

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
```

Vous pouvez également supprimer l'ancien système de fichiers si vous le souhaitez. 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 235](#)
- [“Réception d'un instantané ZFS” à la page 236](#)
- [“Envoi et réception de flux d'instantanés ZFS complexes” à la page 237](#)
- [“Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde” à la page 235](#)

Les solutions de sauvegarde suivantes sont disponibles pour enregistrer les données ZFS :

- **Produits de sauvegarde d'entreprise** : si vous souhaitez disposer des fonctions suivantes, considérez une solution de sauvegarde d'entreprise :
 - Restauration fichier par fichier
 - Vérification des médias de sauvegarde
 - Gestion des médias
- **Instantanés de systèmes de fichiers et restauration d'instantanés** : exécutez les commandes `zfs snapshot` et `zfs rollback` pour créer facilement une copie d'un système de fichiers et restaurer une version précédente d'un système de fichier, le cas échéant. Par exemple, vous pouvez utiliser cette solution pour restaurer un ou plusieurs fichiers issus d'une version précédente d'un système de fichiers.

Pour de plus amples informations sur la création et la restauration d'instantané, reportez-vous à la section [“Présentation des instantanés ZFS” à la page 225](#).

- **Enregistrement d'instantanés** : utilisez les commandes `zfs send` et `zfs receive` pour envoyer et recevoir un instantané ZFS. Vous pouvez enregistrer les modifications incrémentielles entre instantanés, mais la restauration individuelle de fichiers est

impossible. L'instantané du système doit être restauré dans son intégralité. Ces commandes ne constituent pas une solution de sauvegarde complète pour l'enregistrement de vos données ZFS.

- **Réplication distante** : utilisez les commandes `zfs send` et `zfs receive` pour copier un système de fichiers d'un système vers un autre. Ce processus diffère d'un produit de gestion de volume classique qui pourrait mettre les périphériques en miroir dans un WAN. Aucune configuration ni aucun matériel spécifique n'est requis. La réplication de systèmes de fichiers ZFS a ceci d'avantageux qu'elle permet de recréer un système de fichiers dans un pool de stockage et de spécifier différents niveaux de configuration pour le nouveau pool, comme RAID-Z, mais avec des données de système de fichiers identiques.
- **Utilitaires d'archivage** : enregistrez les données ZFS à l'aide d'utilitaires d'archivage tels que `tar`, `cpio` et `pax`, ou des produits de sauvegarde tiers. Actuellement, les deux utilitaires `tar` et `cpio` traduisent correctement les ACL de type NFSv4, contrairement à l'utilitaire `pax`.

Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde

Outre les commandes `zfs send` et `zfs receive`, vous pouvez utiliser des utilitaires d'archivage, tels que les commandes `tar` et `cpio` pour enregistrer des fichiers ZFS. Ces utilitaires enregistrent et restaurent les attributs de fichiers et les ACL ZFS. Vérifiez les options adéquates des commandes `tar` et `cpio`.

Pour connaître les toutes dernières informations relatives à ZFS et aux produits de sauvegarde tiers, consultez les notes de version de Solaris 10 ou la FAQ de ZFS à l'adresse :

<http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq/#backupsoftware>

Envoi d'un instantané ZFS

Vous pouvez utiliser la commande `zfs send` pour envoyer une copie d'un flux d'instantané et recevoir ce flux dans un autre pool du même système ou dans un autre pool d'un système différent utilisé pour stocker les données de sauvegarde. Par exemple, pour envoyer le flux d'instantané à un pool différent du même système, employez une syntaxe du type suivant :

```
# zfs send tank/data@snap1 | zfs recv spool/ds01
```

Vous pouvez utiliser `zfs recv` en tant qu'alias pour la commande `zfs receive`.

Si vous envoyez le flux de l'instantané à un système différent, envoyez la sortie de la commande `zfs send` à la commande `ssh`. 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 (`snap1`) correspond à l'instantané le plus ancien, le second (`snap2`) à l'instantané le plus récent. Dans ce cas, le système de fichiers `newtank/dana` doit déjà exister pour que la réception incrémentielle s'effectue correctement.

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 de raccourci est équivalente à la syntaxe incrémentielle de l'exemple précédent.

Le message s'affiche en cas de tentative de génération d'un flux incrémentiel à partir d'un instantané1 provenant d'un autre système de fichiers :

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Si vous devez stocker de nombreuses copies, envisagez de compresser une représentation de flux d'instantané ZFS à l'aide de la commande `gzip`. Exemple :

```
# zfs send pool/fs@snap | gzip > backupfile.gz
```

Réception d'un instantané ZFS

Gardez les points suivants à l'esprit lorsque vous recevez un instantané d'un système de fichiers :

- L'instantané et le système de fichiers sont reçus.
- Le système de fichiers et tous les systèmes de fichiers descendants sont démontés.
- Les systèmes de fichiers sont inaccessibles tant qu'ils sont en cours de réception.
- Le système de fichiers d'origine à recevoir ne doit pas exister tant qu'il est en cours de transfert.
- Si ce nom existe déjà, vous pouvez utiliser la commande `zfs rename` pour renommer le système de fichiers.

Exemple :

```
# zfs send tank/gozer@0830 > /bkups/gozer.083006
# zfs receive tank/gozer2@today < /bkups/gozer.083006
# zfs rename tank/gozer tank/gozer.old
# zfs rename tank/gozer2 tank/gozer
```

Si vous apportez des modifications au système de fichiers de destination et souhaitez effectuer un autre envoi incrémentiel d'instantané, vous devez au préalable restaurer le système de fichiers destinataire.

Voyez l'exemple suivant : Modifiez tout d'abord le système de fichiers comme suit :

```
host2# rm newtank/dana/file.1
```

Effectuez ensuite un envoi incrémentiel de `char/dana@snap3`. Cependant, vous devez d'abord annuler (roll back) le système de fichiers destinataire pour permettre la réception du nouvel instantané incrémentiel. Vous pouvez aussi utiliser l'option `-F` pour éviter l'étape de restauration. 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`, un message similaire au 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 dans la commande `zfs receive`, la restauration et la réception ne s'effectuent pas correctement et le message d'erreur suivant s'affiche :

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Envoi et réception de flux d'instantanés ZFS complexes

Cette section décrit l'utilisation des options `zfs send -I` et `-R` pour envoyer et recevoir des flux d'instantanés plus complexes.

Gardez les points suivants à l'esprit lors de l'envoi et de la réception de flux d'instantanés ZFS complexes :

- Utilisez l'option `zfs send -I` pour envoyer tous les flux incrémentiels d'un instantané à un instantané cumulé. Vous pouvez également utiliser cette option pour envoyer un flux incrémentiel de l'instantané d'origine pour créer un clone. L'instantané d'origine doit déjà exister sur le côté récepteur afin d'accepter le flux incrémentiel.
- Utilisez l'option `zfs send -R` pour envoyer un flux de réplication de tous les systèmes de fichiers descendants. Une fois le flux de réplication reçu, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.
- Vous pouvez utiliser les deux options pour envoyer un flux de réplication incrémentiel.
 - Les modifications des propriétés sont conservées, tout comme les opérations `rename` et `dest roy` des instantanés et des systèmes de fichiers.
 - Si l'option `zfs recv -F` n'est pas spécifiée lors de la réception du flux de réplication, les opérations `dest roy` du 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 - `i` ou `-I` (autres que `zfs send -R`), si l'option `-I` est utilisée, tous les instantanés créés entre `snapA` et `snapD` sont envoyés. Si l'option `-i` est utilisée, seul `snapD` (pour tous les descendants) est envoyé.
- Pour recevoir ces nouveaux types de flux `zfs send`, le système récepteur doit exécuter une version du logiciel capable de les envoyer. La version des flux est incrémentée.

Vous pouvez cependant accéder à des flux d'anciennes versions de pool en utilisant une version plus récente du logiciel. Vous pouvez par exemple envoyer et recevoir des flux créés à l'aide des nouvelles options à partir d'un pool de la version 3. Vous devez par contre exécuter un logiciel récent pour recevoir un flux envoyé avec les nouvelles options.

EXEMPLE 7-1 Envoi et réception de flux d'instantanés ZFS complexes

Plusieurs instantanés incrémentiels peuvent être regroupés en un seul instantané à l'aide de l'option `zfs send -I`. Exemple :

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

Vous pouvez ensuite supprimer `snapB`, `snapC` et `snapD`.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Pour recevoir les instantanés combinés, vous devez utiliser la commande suivante :

```
# zfs receive -d -F pool/fs < /snaps/fs@all-I
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
pool                 428K  16.5G   20K    /pool
pool/fs              71K   16.5G   21K    /pool/fs
```

EXEMPLE 7-1 Envoi et réception de flux d'instantanés ZFS complexes (Suite)

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

Vous pouvez utiliser la commande `zfs send -R` pour répliquer un système de fichiers ZFS et tous ses systèmes de fichiers descendants, jusqu'à l'instantané nommé. Une fois ce flux reçu, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.

Dans l'exemple suivant, des instantanés des systèmes de fichiers utilisateur sont créés. Un flux de réplication de tous les instantanés utilisateur est créé. Les systèmes de fichiers et instantanés d'origine sont ensuite détruits et récupérés.

```
# zfs snapshot -r users@today
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
users                187K  33.2G  22K    /users
users@today          0      -      22K    -
users/user1          18K   33.2G  18K    /users/user1
users/user1@today    0      -      18K    -
users/user2          18K   33.2G  18K    /users/user2
users/user2@today    0      -      18K    -
users/user3          18K   33.2G  18K    /users/user3
users/user3@today    0      -      18K    -
# zfs send -R users@today > /snaps/users-R
# zfs destroy -r users
# zfs receive -F -d users < /snaps/users-R
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
users                196K  33.2G  22K    /users
users@today          0      -      22K    -
users/user1          18K   33.2G  18K    /users/user1
users/user1@today    0      -      18K    -
users/user2          18K   33.2G  18K    /users/user2
users/user2@today    0      -      18K    -
users/user3          18K   33.2G  18K    /users/user3
users/user3@today    0      -      18K    -
```

Dans l'exemple suivant, la commande `zfs send -R` a été utilisée pour répliquer le jeu de données `users` et ses descendants, puis envoyer le flux répliqué à un autre pool, `users2`.

EXEMPLE 7-1 Envoi et réception de flux d'instantanés ZFS complexes (Suite)

```
# zfs create users2 mirror c0t1d0 c1t1d0
# zfs receive -F -d users2 < /snaps/users-R
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
users	224K	33.2G	22K	/users
users@today	0	-	22K	-
users/user1	33K	33.2G	18K	/users/user1
users/user1@today	15K	-	18K	-
users/user2	18K	33.2G	18K	/users/user2
users/user2@today	0	-	18K	-
users/user3	18K	33.2G	18K	/users/user3
users/user3@today	0	-	18K	-
users2	188K	16.5G	22K	/users2
users2@today	0	-	22K	-
users2/user1	18K	16.5G	18K	/users2/user1
users2/user1@today	0	-	18K	-
users2/user2	18K	16.5G	18K	/users2/user2
users2/user2@today	0	-	18K	-
users2/user3	18K	16.5G	18K	/users2/user3
users2/user3@today	0	-	18K	-

Réplication distante de données ZFS

Les commandes `zfs send` et `zfs recv` permettent d'effectuer une copie distante d'une représentation de flux d'instantané d'un système vers un autre. Exemple :

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Cette commande envoie les données de l'instantané `tank/cindy@today` et les reçoit dans le système de fichiers `sandbox/restfs`. La commande suivante crée également un instantané `restfs@aujourd'hui` sur le système `newsys`. Dans cet exemple, l'utilisateur a été configuré pour utiliser `ssh` dans le système distant.

Utilisation des ACL pour protéger les fichiers Oracle Solaris ZFS

Ce chapitre explique comment utiliser des ACL (Access Control List) pour protéger les fichiers ZFS. Les ACL octroient des droits d'accès plus granulaires que les droits d'accès UNIX standard.

Il contient les sections suivantes :

- “Nouveau modèle ACL Solaris” à la page 241
- “Configuration d'ACL dans des fichiers ZFS” à la page 248
- “Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé” à la page 250
- “Configuration et affichage d'ACL dans des fichiers ZFS en format compact” à la page 262

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 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 principales différences du nouveau modèle ACL sont les suivantes :

- Le nouveau modèle ACL est basé sur la spécifications NFSv4 et est similaire aux ACL de type Windows NT.
- Ce nouveau modèle fournit un jeu d'autorisations d'accès plus détaillé. Pour plus d'informations, reportez-vous au [Tableau 8-2](#).
- Les ACL sont définies et affichées avec les commandes `chmod` et `ls` plutôt qu'avec les commandes `setfacl` et `getfacl`.

- Le nouveau modèle propose une sémantique héritée plus riche pour définir l'application des privilèges d'accès du répertoire aux sous-répertoires, etc. Pour plus d'informations, reportez-vous à la section [“Héritage d'ACL” à la page 246](#).

Les deux modèles d'ACL assurent un contrôle d'accès 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 permissions autorisées 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. Par conséquent, il est impossible de déduire de toute entrée de contrôle d'accès (ACE, Access Control Entry) 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.
- Certaines ACL NFSv4 sont converties en ACL POSIX. Si une ACL NFSv4 n'a pas été convertie en ACL POSIX, un message tel que le suivant s'affiche :

```
# 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 contiennent des ACL POSIX, ces dernières sont converties en ACL NFSv4.
- Si vous tentez de définir une ACL NFSv4 dans un fichier UFS, un message tel que le suivant s'affiche :

```
chmod: ERROR: ACL type's are different
```

- Si vous tentez de définir une ACL POSIX dans un fichier UFS, des messages tels que les suivants s'affichent :

```
# 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 235](#).

Descriptions de syntaxe pour la configuration des ACL

Les deux formats de base des ACL sont les suivants :

Syntax for Setting Trivial ACLs

Une ACL constitue un élément *trivial* : elle représente uniquement les entrées UNIX owner/group/other standard.

```
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'entrée d'ACL, reportez-vous au [Tableau 8-1](#).

```
user ou group: ACL-entry-ID=username or groupname
```

Identifie le *type d'entrée d'ACL* pour la syntaxe d'ACL explicite. Les *types d'entrées d'ACL* user et group doivent également contenir les éléments suivants : *ACL-entry-ID*, *username* ou *groupname*. Pour obtenir une description des types d'entrée 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
```

Dans l'exemple suivant, la valeur **2**, à savoir la désignation de la propriété *ID d'index*, permet d'identifier l'entrée ACL dans la plus grande ACL qui peut présenter plusieurs entrées pour le propriétaire, des UID spécifiques, des groupes 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 de l'index 3 A3 en utilisant une syntaxe de commande `chmod` semblable à la syntaxe suivante :

```
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. Cette entrée doit comprendre l' <i>ID d'entrée ACL</i> contenant un <i>nom d'utilisateur</i> ou un <i>ID d'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. Cette entrée doit comprendre l' <i>ID d'entrée ACL</i> contenant 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.
execute	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 les informations d'un fichier.
read_xattr	R	Droit de lire les attributs d'un fichier ou d'effectuer une recherche dans le répertoire d'attributs étendus d'un fichier.
synchronize	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 horodatages associés à fichier ou un répertoire par une valeur arbitraire.
write_acl	C	Droit d'écriture sur l'ACL ou de modification de celle-ci 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 d'hériter des ACL qui lui sont destinées, tout en tenant compte des droits existants du répertoire parent.

Par défaut, les ACL ne sont pas propagées. Si vous définissez une ACL non triviale sur un répertoire, les répertoires enfant n'en héritent pas. 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 à partir du répertoire parent mais s'applique uniquement aux fichiers du répertoire.
dir_inherit	d	Hérite de l'ACL à partir du répertoire parent mais s'applique uniquement au 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.

Propriétés ACL

Un système de fichiers ZFS contient deux propriétés en rapport avec les 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 nouveau fichier ou répertoire est égale aux permissions du fichier ou répertoire.
 - `noallow` – Pour les nouveaux objets, seules les entrées d'ACL héritables dont le type d'accès est `deny` sont héritées.
 - `restricted` – Pour les nouveaux objets, les droits `write_owner` et `write_acl` sont supprimés lorsqu'une entrée d'ACL est héritée.
 - `passthrough` – Lorsque la valeur de la propriété est définie sur `passthrough`, les fichiers sont créés avec les permissions déterminées par les ACE héritables. Si aucune ACE pouvant être héritée n'affecte les permissions, celles-ci sont alors définies en fonction des permissions demandées à partir de l'application.
 - `passthrough-x` – Cette valeur de propriété a la même sémantique que `passthrough`, si ce n'est que lorsque `passthrough-x` est activé, les fichiers sont créés avec la permission d'exécution (`x`), mais uniquement si la permission d'exécution est définie en mode de création de fichier et dans une ACE pouvant être héritée et qui affecte le mode.

La valeur par défaut de la propriété `aclinherit` est `restricted`.

- `aclmode` – Cette propriété modifie le comportement des ACL lorsqu'un fichier est créé ou chaque fois que les permissions d'un fichier ou d'un répertoire sont modifiées 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 permissions d'ACL de groupe ou d'utilisateur sont réduites de manière à ce qu'elles ne soient pas supérieures aux permissions 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 permissions d'ACL sont réduites afin qu'elles ne dépassent pas les permissions 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`.

La valeur 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'entrées d'ACL. ZFS fournit un modèle ACL *pur* dans lequel tous les fichiers disposent d'une ACL. L'ACL est habituellement *triviale* dans la mesure où elle ne représente que les entrées UNIX traditionnelles `owner/group/other`.

Si vous modifiez les permissions 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 permission 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 principales règles d'accès aux ACL dans un fichier ZFS sont les suivantes :

- ZFS les entrées d'ACL suivant l'ordre dans lesquelles elles sont répertoriées dans l'ACL, de haut en bas.
- Seules les entrées d'ACL disposant d'un " who " correspondant au demandeur d'accès sont traitées.
- Une fois la permission `allow` accordée, celle-ci ne peut plus être refusée par une entrée d'ACL `deny` du même jeu de permissions d'ACL.
- Le propriétaire d'un fichier dispose de la permission `write_acl` de façon inconditionnelle, même si celle-ci est explicitement refusée. Dans le cas contraire, tout droit non spécifié est refusé.

Dans les cas de permissions `deny` ou lorsqu'une permission d'accès à un fichier est manquante, 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 ne puissent plus 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 255.

Lorsque vous créez un fichier, en fonction de la valeur `umask`, une ACL triviale par défaut, similaire à la suivante, est appliquée :

```
$ ls -v file.1
-rw-r--r--  1 root    root      206663 May 20 14:09 file.1
0:owner@:execute:deny
```



```

1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow

```

Notez que chaque catégorie d'utilisateur (owner@, group@, everyone@) dispose de deux entrées ACL dans cet exemple. Une entrée correspond aux permissions deny et une autre aux permissions allow.

La description de l'ACL de ce fichier est la suivante :

- 0:owner@ Le propriétaire ne dispose pas de la permission d'exécution sur le fichier (execute:deny).
- 1:owner@ Le propriétaire peut lire et modifier le contenu d'un fichier (read_data/write_data/append_data). Le propriétaire peut également modifier les attributs du fichier, tels que les horodatages, les attributs étendus et les ACL (write_xattr/write_attributes /write_acl). Le propriétaire peut également modifier la propriété du fichier (write_owner:allow).
- 2:group@ Les permissions de modification et d'exécution sur fichier sont refusées 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@ Tout utilisateur n'étant pas le propriétaire du fichier ou membre du groupe propriétaire du fichier, ne peut ni exécuter, ni modifier les informations ou les attributs d'un fichier (write_data/append_data/write_xattr/execute/write_attributes/write_acl/write_owner:deny).
- 5:everyone@ Tout utilisateur n'étant pas le propriétaire du fichier ou membre du groupe propriétaire du fichier dispose de permissions de lecture sur le fichier et ses attributs (read_data/read_xattr/read_attributes/read_acl/synchronize:allow). Le droit d'accès synchronize n'est actuellement pas implémenté.

Lorsque vous créez un répertoire dépendant de la valeur umask, un répertoire ACL par défaut, semblable au répertoire ACL suivant, est appliqué :

```

$ ls -dv dir.1
drwxr-xr-x 2 root root 2 May 20 14:11 dir.1
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr

```

```

/write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

La description de l'ACL de ce répertoire est la suivante :

- 0:owner@ La liste deny 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 répertoire, 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 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@ Tout utilisateur n'étant pas le propriétaire du fichier ou membre du groupe propriétaire du fichier ne peut ni ajouter, ni modifier le contenu du répertoire (add_file/write_data/add_subdirectory/append_data). En outre, cet utilisateur ne dispose pas de la permission de modifier les attributs du répertoire (write_xattr/write_attributes/write_acl/write_owner:refuser).
- 5:everyone@ Tout utilisateur n'étant pas le propriétaire du fichier ou membre du groupe propriétaire du fichier dispose de permissions de lecture et d'exécution sur le contenu et les attributs du répertoire (list_directory/les permissions read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow). Le droit d'accès synchronize 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 obtenir une description de la propriété *acl-specification*, reportez-vous à la section “[Descriptions de syntaxe pour la configuration des ACL](#)” à la page 243.

- Ajout d'entrées d'ACL
 - Ajout d'une entrée d'ACL pour un utilisateur
 - % `chmod A+acl-specification filename`
 - Ajout d'une entrée d'ACL par *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 May 20 14:09 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
   /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
   /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
   :allow
```

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 262.

EXEMPLE 8-1 Modification des ACL triviales dans des fichiers ZFS

Cette section contient des exemples de définition et d'affichage des ACL triviales.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.1` :

```
# ls -v file.1
-rw-r--r--  1 root   root       206663 May 20 15:03 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
```

EXEMPLE 8-1 Modification des ACL triviales dans des fichiers ZFS (Suite)

```

/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 permissions `write_data` sont accordées 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 20 15:03 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
/write_acl/write_owner:allow
 2:group@: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 permissions du fichier `file.1` sont reconfigurées sur 644.

```

# chmod 644 file.1
# ls -v file.1
-rw-r--r-- 1 root    root      206663 May 20 15:03 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
/write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
/write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow

```

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, l'utilisateur `gozer` dispose de des droits d'accès `read_data/execute` sur 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 May 20 15:09 test.dir
 0:user:gozer:list_directory/read_data/execute:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory

```

EXEMPLE 8-2 Configuration d'ACL non triviales dans des fichiers ZFS (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 permissions `read_data/execute` sont retirées à l'utilisateur `gozer` :

```

# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x  2 root    root          2 May 20 15:09 test.dir
0:owner@::deny
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 illustrent l'interaction entre la configuration d'ACL et les modifications des permissions du fichier ou du répertoire.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.2`:

```

# ls -v file.2
-rw-r--r--  1 root    root          3103 May 20 15:23 file.2
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow

```

Dans l'exemple suivant, les permissions d'ACL (`allow`) sont retirées à `everyone@`:

```

# chmod A5- file.2
# ls -v file.2
-rw-r-----  1 root    root          3103 May 20 15:23 file.2
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow

```

EXEMPLE 8-3 Interactions entre les ACL et les droits dans les fichiers ZFS (Suite)

```

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 permissions du fichier sont réinitialisées de 644 à 640. Les permissions de lecture pour `everyone@` sont supprimées des permissions du fichier lorsque les permissions d'ACL (`allow`) sont retirées à `everyone@`.

Dans l'exemple suivant, l'ACL existante est remplacée par des permissions `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        6986 May 20 15:25 file.3
  0:everyone@:read_data/write_data:allow

```

Dans cette sortie, la syntaxe `chmod` remplace effectivement l'ACL existante par les permissions `read_data/write_data:allow` pour les permissions 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 permissions du propriétaire ou du groupe, les permissions sont définies sur 666.

Dans l'exemple suivant, l'ACL existante est remplacée par des permissions de lecture pour l'utilisateur `gozer`:

```

# chmod A=user:gozer:read_data:allow file.3
# ls -v file.3
-----+ 1 root    root        6986 May 20 15:25 file.3
  0:user:gozer:read_data:allow

```

Dans cette sortie, les permissions de fichier sont calculées pour être `000`, car aucune entrée d'ACL n'existe pour `owner@`, `group@` ou `everyone@`, qui représentent les composants de permission 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        6986 May 20 15:25 file.3
  0:user:gozer::deny
  1:user:gozer:read_data:allow
  2:owner@:execute:deny
  3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:allow
  4:group@:write_data/append_data:deny
  5:group@:read_data/execute:allow
  6:everyone@:write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:deny
  7:everyone@:read_data/read_xattr/execute/read_attributes/read_acl

```

EXEMPLE 8-3 Interactions entre les ACL et les droits dans les fichiers ZFS (Suite)

```
/synchronize:allow
```

EXEMPLE 8-4 Restauration des ACL triviales dans des fichiers ZFS

Vous pouvez utiliser la commande `chmod` pour supprimer les ACL non triviales d'un fichier ou d'un répertoire, ce qui permet de restaurer les ACL triviales du fichier ou du répertoire.

Dans l'exemple suivant, deux ACE non triviales existent dans `test5.dir` :

```
# ls -dv test5.dir
drwxr-xr-x+ 2 root    root          2 May 20 15:32 test5.dir
0:user:lp:read_data:file_inherit:deny
1:user:gozer:read_data:file_inherit:deny
2:owner@::deny
3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /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
```

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 May 20 15:32 test5.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé

Vous pouvez spécifier les conditions et la façon dont les ACL sont héritées dans les fichiers et les 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 `restricted` et `aclmode` sur `groupmask`.

Pour plus d'informations, reportez-vous à la section “[Héritage d'ACL](#)” à la page 246.

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 triviale de `read_data/write_data/execute` est appliquée pour l'utilisateur `gozer` dans le répertoire `test.dir` :

```
# chmod A+user:gozer:read_data/write_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+ 2 root    root          2 May 20 15:41 test.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Si un sous-répertoire `test.dir` est créé, l'ACE pour l'utilisateur `gozer` n'est pas propagée. L'utilisateur `gozer` n'aurait accès au sous-répertoire que si les permissions du sous-répertoire lui accordaient un accès en tant que propriétaire de fichier, membre d'un groupe ou comme faisant partie de la propriété `everyone@`. Exemple :

```
# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x 2 root    root          2 May 20 15:42 test.dir/sub.dir
 0:owner@::deny
 1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 2:group@:add_file/write_data/add_subdirectory/append_data:deny
 3:group@:list_directory/read_data/execute:allow
 4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires

Les exemples suivants permettent d'identifier les ACE du fichier et du répertoire qui son appliquées lorsque l'indicateur `file_inherit` est paramétré.

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

Dans cet exemple, les permissions `read_data/write_data` sont ajoutées pour les fichiers dans le répertoire `test2.dir` pour l'utilisateur `gozer` pour qu'il dispose de l'accès en lecture à tout nouveau fichier :

```
# chmod A+user:gozer:read_data/write_data:file_inherit:allow test2.dir
# ls -dv test2.dir
drwxr-xr-x+ 2 root      root          2 May 20 15:50 test2.dir
 0:user:gozer:read_data/write_data:file_inherit:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 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 permissions de l'utilisateur `gozer` sont appliquées au fichier `test2.dir/file.2` récemment créé. L'héritage d'ACL étant accordé (`read_data:file_inherit:allow`), l'utilisateur `gozer` peut lire le contenu de tout nouveau fichier.

```
# touch test2.dir/file.2
# ls -v test2.dir/file.2
-rw-r--r--+ 1 root      root          0 May 20 15:51 test2.dir/file.2
 0:user:gozer:write_data:deny
 1:user:gozer:read_data/write_data:allow
 2:owner@:execute:deny
 3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
 4:group@:write_data/append_data/execute:deny
 5:group@:read_data:allow
 6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
 7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

Étant donné que la propriété `aclmode` de ce fichier est paramétrée sur la valeur par défaut `groupmask`, l'utilisateur `gozer` ne dispose pas de la permission `write_data` pour le fichier `file.2` car la permission de groupe du fichier ne le permet pas.

La permission `inherit_only` appliquée lorsque les indicateurs `file_inherit` ou `dir_inherit` sont définis, est utilisées pour propager l'ACL dans la structure du répertoire. Ainsi, l'utilisateur `gozer` se voit uniquement accorder ou refuser la permission des droits `everyone@`, à moins qu'il ne soit le propriétaire du fichier ou membre du groupe propriétaire du fichier. Exemple :

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```
# mkdir test2.dir/subdir.2
# ls -dv test2.dir/subdir.2
drwxr-xr-x+ 2 root    root          2 May 20 15:52 test2.dir/subdir.2
 0:user:gozer:list_directory/read_data/add_file/write_data:file_inherit
  /inherit_only:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Les exemples suivants identifient les ACL du fichier et du répertoires appliquées, lorsque les indicateurs `file_inherit` et `dir_inherit` sont paramétrés.

Dans cet exemple, l'utilisateur `gozer` se voit accorder les permissions de lecture, d'écriture et d'exécution héritées 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 May 20 15:53 test3.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
  :file_inherit/dir_inherit:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

# touch test3.dir/file.3
# ls -v test3.dir/file.3
-rw-r--r--+ 1 root    root          0 May 20 15:58 test3.dir/file.3
 0:user:gozer:write_data/execute:deny
 1:user:gozer:read_data/write_data/execute:allow
 2:owner@:execute:deny
 3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
 4:group@:write_data/append_data/execute:deny
 5:group@:read_data:allow
 6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
 7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```
# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+ 2 root    root          2 May 20 15:59 test3.dir/subdir.1
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
   :file_inherit/dir_inherit/inherit_only:allow
 1:user:gozer:add_file/write_data:deny
 2:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
 3:owner@::deny
 4:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
   /append_data/write_xattr/execute/write_attributes/write_acl
   /write_owner:allow
 5:group@:add_file/write_data/add_subdirectory/append_data:deny
 6:group@:list_directory/read_data/execute:allow
 7:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
   /write_attributes/write_acl/write_owner:deny
 8:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
   /read_acl/synchronize:allow
```

Dans ces exemples, les permissions du répertoire parent pour `group@` et `everyone@` n'accordent pas les permissions d'écriture et d'exécution. Par conséquent, l'utilisateur `gozer` se voit refuser ces permissions. La propriété par défaut `aclinherit` est de `restricted`, ce qui signifie que les permissions `write_data` et `execute` ne sont pas héritées.

Dans l'exemple suivant, l'utilisateur `gozer` se voit accorder les permissions de lecture, d'écriture et d'exécution héritées des fichiers récemment créés. Toutefois, elles ne sont pas propagées vers le 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 May 20 16:02 test4.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
   :file_inherit/no_propagate:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
   /append_data/write_xattr/execute/write_attributes/write_acl
   /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
   /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
   /read_acl/synchronize:allow
```

Comme l'exemple suivant l'illustre, lors de la création d'un sous-répertoire, la permission `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 May 20 16:03 test4.dir/sub4.dir
```

EXEMPLE 8-6 Attribution d'héritage d'ACL dans les fichiers et les répertoires (Suite)

```

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, les permissions `read_data/write_data/execute` de l'utilisateur `gozer` pour les fichiers sont propagées vers le nouveau fichier :

```

# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 May 20 16:04 test4.dir/file.4
0:user:gozer:write_data/execute:deny
1:user:gozer:read_data/write_data/execute:allow
2:owner@:execute:deny
3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
4:group@:write_data/append_data/execute:deny
5:group@:read_data:allow
6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow

```

EXEMPLE 8-7 Héritage d'ACL avec la propriété `aclmode` définie sur `passthrough`

Comme le montre l'exemple suivant, lorsque la propriété `aclmode` du système de fichiers `tank/cindys` est définie sur `passthrough`, l'utilisateur `gozer` hérite de l'ACL appliquée au répertoire `test4.dir` du fichier `file.4` récemment créé :

```

# zfs set aclmode=passthrough tank/cindys
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 May 20 16:08 test4.dir/file.4
0:user:gozer:write_data/execute:deny
1:user:gozer:read_data/write_data/execute:allow
2:owner@:execute:deny
3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
4:group@:write_data/append_data/execute:deny
5:group@:read_data:allow
6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow

```

EXEMPLE 8-7 Héritage d'ACL avec la propriété `aclmode` définie sur `passthrough` (Suite)

Cette sortie indique 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 la propriété `aclmode` définie sur `discard`

Si la propriété `aclmode` d'un système de fichiers est définie sur `discard`, il est alors possible de supprimer les ACL en cas de modification des permissions dans un répertoire. Exemple :

```
# zfs set aclmode=discard tank/cindys
# chmod A+user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
# ls -dv test5.dir
drwxr-xr-x+ 2 root      root          2 May 20 16:09 test5.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
   :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 par la suite de renforcer les permissions d'un répertoire, l'ACL non triviale est supprimée. Exemple :

```
# chmod 744 test5.dir
# ls -dv test5.dir
drwxr--r-- 2 root      root          2 May 20 16:09 test5.dir
 0:owner@::deny
 1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
   /append_data/write_xattr/execute/write_attributes/write_acl
   /write_owner:allow
 2:group@:add_file/write_data/add_subdirectory/append_data/execute:deny
 3:group@:list_directory/read_data:allow
 4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
   /execute/write_attributes/write_acl/write_owner:deny
 5:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
   /synchronize:allow
```

EXEMPLE 8-9 Héritage d'ACL avec la propriété `aclinherit` définie 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. L'exemple suivant montre également comment spécifier deux ACE dans la même commande `chmod`.

EXEMPLE 8-9 Héritage d'ACL avec la propriété `aclinherit` définie sur `noallow` (Suite)

```
# zfs set aclinherit=noallow tank/cindys
# chmod A+user:gozer:read_data:file_inherit:deny,user:lp:read_data:file_inherit:allow
test6.dir
# ls -dv test6.dir
drwxr-xr-x+ 2 root    root          2 May 20 16:11 test6.dir
 0:user:gozer:read_data:file_inherit:deny
 1:user:lp:read_data:file_inherit:allow
 2:owner@::deny
 3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 4:group@:add_file/write_data/add_subdirectory/append_data:deny
 5:group@:list_directory/read_data/execute:allow
 6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 7:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

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 May 20 16:13 test6.dir/file.6
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

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 permissions compactes 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 Jun 17 10:07 file.1
  owner@:--x-----:-----:deny
  owner@:rw-p---A-W-Co-:-----:allow
```

```

group@: -w xp-----:-----: deny
group@: r-----:-----: allow
everyone@: -w xp---A-W-Co-:-----: deny
everyone@: r-----a-R-c--s:-----: allow

```

Cette sortie d'ACL compacte est décrite comme suit :

owner@	Les permissions d'exécution sur le fichier est refusée au propriétaire (x=execute).
owner@	Le propriétaire peut lire ou modifier le contenu du fichier (rw=read_data/write_data, p=append_data). Il peut également modifier les attributs du fichier, tels que les horodatages, les attributs étendus et les ACL (A=write_xattr, W=write_attributes et 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 permissions de lecture sur le fichier sont accordées au groupe (r=read_data).
everyone@	Les permissions 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 son propriétaire ni membre du groupe propriétaire du (w= write_data, x=execute, p =append_data, A=write_xattr, W=write_attributes, C= write_acl et o=write_owner).
everyone@	Toute personne n'étant pas le propriétaire du fichier ou membre du groupe propriétaire du fichier ne peut ni lire, ni synchroniser le fichier et ses attributs (r= read_data, a=append_data, R=read_xattr, c=read_acl et s=synchronize). Le droit d'accès synchronize n'est actuellement pas implémenté.

Le format d'ACL compact dispose des avantages suivants par rapport au format détaillé :

- Les droits peuvent être spécifiés en tant qu'arguments de position pour la commande `chmod`.
- Les traits d'union (-) permettant d'identifier l'absence de permission 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 250.

EXEMPLE 8-10 Configuration et affichage des ACL en format compact

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.1` :

EXEMPLE 8-10 Configuration et affichage des ACL en format compact (Suite)

```
# ls -V file.1
-rw-r--r-- 1 root    root      206663 Jun 17 10:07 file.1
  owner@: --x-----:-----:deny
  owner@: rw-p---A-W-Co:-----:allow
  group@: -wxp-----:-----:deny
  group@: r-----:-----:allow
  everyone@: -wpx---A-W-Co:-----:deny
  everyone@: r-----a-R-c--s:-----:allow
```

Dans l'exemple suivant, les permissions `read_data/execute` sont ajoutées à l'utilisateur `gozer` dans le fichier .

```
# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+ 1 root    root      206663 Jun 17 10:07 file.1
  user:gozer:r-x-----:-----:allow
  owner@: --x-----:-----:deny
  owner@: rw-p---A-W-Co:-----:allow
  group@: -wxp-----:-----:deny
  group@: r-----:-----:allow
  everyone@: -wpx---A-W-Co:-----:deny
  everyone@: r-----a-R-c--s:-----:allow
```

Une autre méthode d'ajout des mêmes permissions pour l'utilisateur `gozer` consiste à insérer une entrée d'ACL à un emplacement spécifique, par exemple 4. Ainsi, les ACL existantes aux emplacements 4-6 sont déplacées vers le bas. Exemple :

```
# chmod A4+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+ 1 root    root      206663 Jun 17 10:16 file.1
  owner@: --x-----:-----:deny
  owner@: rw-p---A-W-Co:-----:allow
  group@: -wxp-----:-----:deny
  group@: r-----:-----:allow
  user:gozer:r-x-----:-----:allow
  everyone@: -wpx---A-W-Co:-----:deny
  everyone@: r-----a-R-c--s:-----:allow
```

Dans l'exemple suivant, l'utilisateur `gozer` se voit accorder les droits de lecture, d'écriture et d'exécution hérités des fichiers et répertoires récemment créés.

```
# chmod A+user:gozer:rwx:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+ 2 root    root      2 Jun 17 10:19 dir.2
  user:gozer:rwx-----:fd---:allow
  owner@: -----:-----:deny
  owner@: rwxp---A-W-Co:-----:allow
  group@: -w-p-----:-----:deny
  group@: r-x-----:-----:allow
  everyone@: -w-p---A-W-Co:-----:deny
  everyone@: r-x---a-R-c--s:-----:allow
```


EXEMPLE 8-10 Configuration et affichage des ACL en format compact (Suite)

Vous pouvez également couper et coller les droits et les indicateurs d'héritage à partir de la sortie `ls -lV` en format `chmod` compact. Par exemple, pour dupliquer les permissions et les indicateurs d'héritage du répertoire `dir.2` de l'utilisateur `gozer` vers le répertoire `dir.2` de l'utilisateur `cindys`, copiez et collez les permissions et les indicateurs d'héritage (`rwX-----:fd---:allow`) dans la commande `chmod` comme suit :

```
# chmod A+user:cindys:rwX-----:fd---:allow dir.2
# ls -lV dir.2
drwxr-xr-x+ 2 root      root          2 Jun 17 10:19 dir.2
      user:cindys:rwX-----:fd---:allow
      user:gozer:rwX-----:fd---:allow
      owner@:-----:-----:deny
      owner@:rwxp--A-W-Co:-----:allow
      group@:-w-p-----:-----:deny
      group@:r-x-----:-----:allow
      everyone@:-w-p--A-W-Co:-----:deny
      everyone@:r-x--a-R-c-s:-----:allow
```

EXEMPLE 8-11 Héritage d'ACL avec la propriété `aclinherit` définie 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 des permissions déterminées par les ACE pouvant être héritées. Si aucune ACE pouvant être héritée n'affecte les permissions, celles-ci sont alors définies en fonction des permissions demandées à partir de l'application.

Les exemples suivants utilisent la syntaxe ACL compacte pour illustrer le processus d'héritage des permissions en définissant la propriété `aclinherit` sur `passthrough`.

Dans cet exemple, une ACL est définie sur le répertoire `test1.dir` pour forcer l'héritage. La syntaxe crée une entrée d'ACL `owner@`, `group@` et `everyone@` pour les fichiers nouvellement créés. Les répertoires nouvellement créés héritent d'une entrée d'ACL `@owner`, `@group` et `@everyone`. En outre, les répertoires héritent de 6 autres ACE qui appliquent les ACE aux fichiers et répertoires 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 -lV test1.dir
drwxrwx---+ 2 root      root          2 Jun 17 10:37 test1.dir
      owner@:rwxpdDaARwCcos:fd---:allow
      group@:rwxp-----:fd---:allow
      everyone@:-----:fd---:allow
```

EXEMPLE 8-11 Héritage d'ACL avec la propriété `aclinherit` définie sur `passthrough` (Suite)

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 Jun 17 10:38 file.1
      owner@: rwxpdDaARWcCos:-----:allow
      group@: rwxp-----:-----:allow
      everyone@:-----:-----:allow
```

Dans cet exemple, un répertoire nouvellement créé hérite à la fois des ACE contrôlant l'accès à ce répertoire et des ACE à appliquer ultérieurement aux enfants de ce répertoire.

```
# mkdir subdir.1
# ls -dV subdir.1
drwxrwx---+ 2 root    root          2 Jun 17 10:39 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.

```
# cd /tank/cindys
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r-- 1 root    root          0 Jun 17 10:40 file.2
      owner@:--x-----:-----:deny
      owner@:rw-p--A-W-Co:-----:allow
      group@:-wxp-----:-----:deny
      group@:r-----:-----:allow
      everyone@:-wpx--A-W-Co:-----:deny
      everyone@:r-----a-R-c-s:-----:allow
```

EXEMPLE 8-12 Héritage d'ACL avec la propriété `aclinherit` définie sur `passthrough-x`

Lorsque la propriété `aclinherit` est définie sur `passthrough-x`, les fichiers sont créés avec la permission d'exécution (x) pour `owner@`, `group@` ou `everyone@`, mais seulement si la permission d'exécution est définie dans le mode de création de fichier et dans une ACE héritable affectant ce mode.

L'exemple suivant explique comment hériter de la permission d'exécution en définissant la propriété `aclinherit` sur `passthrough-x`.

EXEMPLE 8-12 Héritage d'ACL avec la propriété `aclinherit` définie sur `passthrough-x` (Suite)

```
# zfs set aclinherit=passthrough-x tank/cindys
```

L'ACL suivante est définie sur `/tank/cindys/test1.dir` pour permettre l'héritage des ACL exécutables pour les fichiers de `owner@`, `group@` et `everyone@`.

```
# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow test1.dir
# ls -Vd test1.dir
drwxrwx---+ 2 root    root          2 Jun 17 10:41 test1.dir
      owner@:rwxpdDaARWcCos:fd----:allow
      group@:rwxp-----:fd----:allow
      everyone@:-----:fd----:allow
```

Un fichier (`file1`) est créé avec les autorisations demandées `0666`. Les autorisations obtenues sont `0660`. L'autorisation d'exécution n'était pas héritée car le mode de création ne le requérait pas.

```
# touch test1.dir/file1
# ls -V test1.dir/file1
-rw-rw----+ 1 root    root          0 Jun 17 10:42 test1.dir/file1
      owner@:rw-pdDaARWcCos:-----:allow
      group@:rw-p-----:-----:allow
      everyone@:-----:-----:allow
```

Ensuite, un fichier exécutable appelé `t` est généré à l'aide du compilateur `cc` dans le répertoire `testdir`.

```
# cc -o t t.c
# ls -V t
-rwxrwx---+ 1 root    root          7396 Jun 17 10:50 t
      owner@:rwxpdDaARWcCos:-----:allow
      group@:rwxp-----:-----:allow
      everyone@:-----:-----:allow
```

Les autorisations obtenues sont `0770` car `cc` a demandé des autorisations `0777`, ce qui a entraîné l'héritage de l'autorisation d'exécution à partir des entrées `propriétaire@`, `groupe@` et tous les `utilisateurs@`.

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 269
- “Délégation de droits ZFS” à la page 270
- “Affichage des droits ZFS délégués (exemples)” à la page 278
- “Délégation de droits ZFS (exemples)” à la page 274
- “Suppression de droits ZFS (exemples)” à la page 279

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 le symbole @ 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. Cette fonctionnalité 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.
- Simplifie l'administration de systèmes de fichiers en réseau (NFS, Network File System). 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

La propriété `delegation` du pool permet de contrôler les fonctions d'administration déléguées. Par exemple :

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

Par défaut, la propriété `delegation` est activée.

Délégation 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
<code>allow</code>	Capacité à accorder des droits qui vous ont été octroyés à un autre utilisateur.	Doit également disposer du droit à autoriser.
<code>Clone</code>	Capacité à cloner tout instantané du jeu de données.	Doit également disposer de la capacité <code>create</code> et de la capacité <code>mount</code> dans le système de fichiers d'origine.
<code>create</code>	Capacité à créer des jeux de données descendants.	Doit également disposer de la capacité <code>mount</code> .
<code>destroy</code>	Capacité à détruire un jeu de données.	Doit également disposer de la capacité <code>mount</code> .
<code>monter</code>	Capacité à monter et démonter un jeu de données, et à créer et détruire les liens vers des périphériques de volume.	
<code>promote</code>	Capacité à promouvoir le clonage d'un jeu de données.	Doit également disposer de la capacité <code>mount</code> et de la capacité <code>promote</code> dans le système de fichiers d'origine.
<code>receive</code>	Capacité à créer des systèmes de fichiers descendant à l'aide de la commande <code>zfs receive</code> .	Doit également disposer de la capacité <code>mount</code> et de la capacité <code>create</code> .
<code>rename</code>	Capacité à renommer un jeu de données.	Doit également disposer de la capacité <code>create</code> et de la capacité <code>mount</code> dans le nouveau parent.
<code>rollback</code>	Capacité à restaurer un instantané.	
<code>send</code>	Capacité à envoyer un flux d'instantané.	
<code>share</code>	Capacité à partager et annuler le partage d'un jeu de données.	
Instantané	Capacité à prendre un instantané de jeu de données.	

Vous pouvez déléguer l'ensemble d'autorisations suivant mais l'autorisation peut être limitée à l'accès, la lecture et la modification :

- `groupquota`
- `groupused`
- `userprop`
- `userquota`
- `userused`

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
- primarycache
- quota
- readonly
- recordsize
- reservation
- reservation
- secondarycache
- setuid
- shareiscsi
- sharenfs
- sharesmb
- snapdir
- utf8only
- version
- volblocksize
- volsize
- vscan
- xattr
- zoned

Certaines de ces propriétés ne peuvent être définies qu'à la création d'un jeu de données. Pour une description de ces propriétés, reportez-vous à la section [“Présentation des propriétés ZFS”](#) à la page 189.

Délégation des autorisations ZFS (`zfs allow`)

La syntaxe de `zfs allow` est la suivante :

```
zfs allow [-ldugecs] everyone|user|group[,...] perm[@setname,...] filesystem| volume
```

La syntaxe de `zfs allow` suivante (en gras) identifie les utilisateurs auxquels les 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 autorisations, 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 de délégation et d'affichage de droits ZFS délégués.

Délégation de droits ZFS (exemples)

EXEMPLE 9-1 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
```

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 9-2 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.

EXEMPLE 9-2 Délégation des droits de création (create) et de destruction (destroy) à un groupe (Suite)

```

# 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 9-3 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 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
-----

```

EXEMPLE 9-3 Délégation de droits au niveau approprié d'un système de fichiers (Suite)

```
# su marks
$ zfs snapshot tank@snap2
cannot create snapshot 'tank@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 9-4 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
# 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 :

EXEMPLE 9-4 Définition et utilisation de droits délégués complexes (Suite)

```

$ cd /tank/marks2
$ ls
$ cd .zfs
$ ls
snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x  2 marks  staff          2 Dec 15 13:53 snap1
$ pwd
/tank/marks2/.zfs/snapshot
$ mkdir snap2
$ zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank                 264K  33.2G  33.5K  /tank
tank/marks           24.5K  33.2G  24.5K  /tank/marks
tank/marks2          46K   33.2G  24.5K  /tank/marks2
tank/marks2@snap1   21.5K  -      24.5K  -
tank/marks2@snap2    0      -      24.5K  -
$ ls
snap1 snap2
$ rmdir snap2
$ ls
snap1

```

EXEMPLE 9-5 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 9-5 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

```

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 9-6 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 9-7 Affichage des droits d'administration déléguée complexes

La sortie de cet exemple indique les droits suivants sur les systèmes de fichiers `pool/fred` et `pool`.

Pour le système de fichiers `pool/fred` :

- Deux jeux de droits sont définis :
 - `@eng` (create, destroy, snapshot, mount, clone, promote, rename)
 - `@simple` (create, mount)

EXEMPLE 9-7 Affichage des droits d'administration déléguée complexes (Suite)

- 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`.
- 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
-----
```

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
-----
```


Rubriques avancées Oracle Solaris 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 281
- “Utilisation de ZFS dans un système Solaris avec zones installées” à la page 284
- “Utilisation de pools racine ZFS de remplacement” à la page 290
- “Profils de droits ZFS” à la page 291

Volumes ZFS

Un volume ZFS est un jeu de données qui représente un périphérique en mode bloc. Les volumes ZFS sont identifiés en tant que périphériques dans le répertoire `/dev/zvol/{dsk, rdsk}/pool`.

Dans l'exemple suivant, un volume ZFS de 5 GO portant le nom `tank/vol` est créé :

```
# zfs create -V 5gb tank/vol
```

Lors de la création d'un volume, une réservation est automatiquement définie sur la taille initiale du volume pour éviter tout comportement inattendu. Si, par exemple, la taille du volume diminue, les données risquent d'être corrompues. Vous devez faire preuve de prudence lors de la modification de la taille du volume.

En outre, si vous créez un instantané pour un volume modifiant la taille de ce dernier, cela peut provoquer des incohérences lorsque vous tentez d'annuler (roll back) l'instantané ou de créer un clone à partir de l'instantané.

Pour de plus amples informations concernant les propriétés de systèmes de fichiers applicables aux volumes, reportez-vous au [Tableau 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. Si vous tentez d'effectuer cette action, cette dernière échouera. Pour obtenir des informations relatives à l'utilisation de volumes ZFS dans une zone globale, reportez-vous à la section [“Ajout de volumes ZFS à une zone non globale” à la page 286.](#)

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 racine UFS, un périphérique de swap est créé sur un volume ZFS du pool racine ZFS. 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 racine UFS, un périphérique de vidage est créé sur un volume ZFS du pool racine ZFS. 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
```

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. Si vous tentez de créer un autre volume de swap, créez un volume ZFS d'une taille spécifique et activez le swap sur le périphérique. Exemple :

```
# zfs create -V 2G rpool/swap2
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile          dev      swaplo  blocks    free
/dev/zvol/dsk/rpool/swap 256,1      16  2097136  2097136
/dev/zvol/dsk/rpool/swap2 256,5      16  4194288  4194288
```

N'effectuez pas de swap vers un fichier dans un système de fichiers ZFS. La configuration de fichier swap ZFS n'est pas prise en charge.

Pour plus d'informations sur l'ajustement de la taille des volumes de swap et de vidage, reportez-vous à la section [“Ajustement de la taille de vos périphériques de swap et de vidage ZFS” à la page 168.](#)

Utilisation d'un volume ZFS en tant que cible iSCSI Solaris

Vous pouvez facilement créer un volume ZFS en tant que cible iSCSI en configurant la propriété `shareiscsi` sur le volume. 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 Oracle 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 Solaris iSCSI. Si vous avez configuré la propriété `shareiscsi` dans un volume ZFS, n'utilisez pas la commande `iscsitadm` pour créer le même périphérique cible. Vous pouvez également dupliquer les informations des cibles sur ce même périphérique.

Vous pouvez gérer un volume ZFS configuré en tant que cible iSCSI de la même façon qu'un autre jeu de données ZFS. Cependant, les opérations de renommage, d'exportation et d'importation fonctionnent de façon différente pour les cibles iSCSI.

- Lors du renommage d'un volume ZFS, le nom de la cible iSCSI ne change pas. 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 d'un système de fichiers ZFS sur un système avec des zones Oracle Solaris :

- “Ajout de systèmes de fichiers ZFS à une zone non globale” à la page 285
- “Délégation de jeux de données à une zone non globale” à la page 286
- “Ajout de volumes ZFS à une zone non globale” à la page 286
- “Utilisation de pools de stockage ZFS au sein d'une zone” à la page 287
- “Gestion de propriétés ZFS au sein d'une zone” à la page 287
- “Explication de la propriété zoned” à la page 288

Pour plus d'informations sur la configuration de zones d'un système de fichiers ZFS racine migré ou auquel des patches ont été appliqués à l'aide de Solaris Live Upgrade, reportez-vous à la section “Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système comportant des zones (Solaris 10 10/08)” à la page 151 ou “Utilisation d'Oracle Solaris Live Upgrade pour migrer ou mettre à jour un système avec zones (version Solaris 5 10/09 ou supérieure)” à la page 156.

Tenez compte des points suivants lors de l'association de jeux de données à des zones :

- Il est possible d'ajouter un système de fichiers ou un clone ZFS à une zone non globale en déléguant ou non le contrôle administratif.
- Vous pouvez ajouter un volume ZFS en tant que périphérique à des zones non globales.
- L'association d'instantanés ZFS à des zones est impossible à l'heure actuelle.

Dans les sections suivantes, le terme jeu de données ZFS fait référence à un système de fichier ou à un clone.

L'ajout d'un jeu de données permet à la zone non globale de partager l'espace avec la zone globale, mais l'administrateur de zone ne peut pas contrôler les propriétés ou créer de nouveaux systèmes de fichiers dans la hiérarchie de systèmes de fichiers sous-jacents. Cette opération est identique à l'ajout de tout autre type de système de fichiers à une zone. Effectuez-la lorsque vous souhaitez simplement partager de l'espace commun.

ZFS autorise également la délégation de jeux de données à une zone non globale, ce qui permet à l'administrateur de zone de contrôler parfaitement le jeu de données et ses enfants.

L'administrateur de zone peut créer et détruire les systèmes de fichiers ou les clones au sein de ce jeu de données et modifier les propriétés des jeux de données. L'administrateur de zone ne peut pas affecter des jeux de données qui n'ont pas été ajoutés à la zone, y compris ceux qui dépassent les quotas de niveau supérieur du jeu de données délégué.

Tenez compte des points suivants lorsque vous utilisez un système de fichiers ZFS dans un système sur lequel des zones Oracle Solaris sont installées :

- La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`.
- En raison du problème 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.
- Lorsqu'un emplacement source `zonepath` et l'emplacement cible `zonepath` résident tous deux dans un système de fichiers ZFS et se trouvent dans le même pool, la commande `zoneadm clone` utilise dorénavant automatiquement le clone ZFS pour cloner une zone. La commande `zoneadm clone` crée un instantané ZFS de la source de l'emplacement `zonepath` et configure l'emplacement `zonepath` cible. Vous ne pouvez pas utiliser la commande `zfs clone` pour cloner une zone. Pour plus d'informations, reportez-vous à la [Partie II, "Zones" du Guide d'administration système : Gestion des ressources des conteneurs et des zones Oracle Solaris](#).
- Si vous déléguez un système de fichiers ZFS à une zone non globale, vous devez supprimer ce système de fichiers de la zone non globale avant d'utiliser Oracle Solaris Live Upgrade. Sinon, l'opération Oracle Solaris Live Upgrade échoue en raison d'une erreur de système de fichier en lecture seule.

Ajout de systèmes de fichiers ZFS à une zone non globale

Vous pouvez ajouter un système de fichiers ZFS en tant que système de fichiers générique lorsqu'il s'agit simplement de partager de l'espace avec la zone globale. La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`.

La sous-commande `add fs` de la commande `zonecfg` permet d'ajouter un système de fichiers ZFS à une zone non globale.

Dans l'exemple suivant, un système de fichiers ZFS est ajouté à une zone non globale par un administrateur global de la zone globale :

```
# zonecfg -z zion
zonecfg:zion> add fs
zonecfg:zion:fs> set type=zfs
zonecfg:zion:fs> set special=tank/zone/zion
zonecfg:zion:fs> set dir=/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 et montée sur `/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 autre emplacement, tout comme l'administrateur ne peut pas modifier les propriétés suivantes du système de fichiers : `atime`,

readonly, compression, etc. L'administrateur de zone globale est chargé de la configuration et du contrôle des propriétés du système de fichiers.

Pour plus d'informations sur la commande `zonecfg` et les types de configuration de la commande `zonecfg`, reportez-vous à la [Partie II, "Zones" du Guide d'administration système : Gestion des ressources des conteneurs et des zones Oracle Solaris](#).

Délégation de jeux de données à une zone non globale

Si l'objectif principal est de déléguer l'administration du stockage d'une zone, le système de fichiers ZFS prend en charge l'ajout de jeux de données à une zone non globale à l'aide de la sous-commande `add dataset` de la commande `zonecfg`.

Dans l'exemple suivant, un système de fichiers ZFS est délégué à une zone non globale par un administrateur global dans la zone globale.

```
# zonecfg -z zion
zonecfg:zion> add dataset
zonecfg:zion:dataset> set name=tank/zone/zion
zonecfg:zion:dataset> end
```

Contrairement à l'ajout d'un système de fichiers, cette syntaxe entraîne la visibilité du système de fichiers ZFS `tank/zone/zion` dans la zone `zion` déjà configurée. L'administrateur de zone peut définir les propriétés de système de fichiers, et créer des systèmes de fichiers descendants. En outre, l'administrateur de zone peut créer des instantanés ainsi que des clones, et contrôler la totalité de la hiérarchie du système de fichiers.

Si vous utilisez Oracle Solaris Live Upgrade pour mettre à niveau l'environnement d'initialisation du système de fichiers ZFS avec des zones non globales, supprimez au préalable tous les jeux de données délégués. Sinon, l'opération Oracle Solaris Live Upgrade échoue en raison d'une erreur de système de fichier en lecture seule. Exemple :

```
zonecfg:zion>
zonecfg:zion> remove dataset name=tank/zone/zion
zonecfg:zion1> exit
```

Pour de plus amples informations relatives aux actions autorisées au sein des zones, reportez-vous à la section ["Gestion de propriétés ZFS au sein d'une zone"](#) à la page 287.

Ajout de volumes ZFS à une zone non globale

Les volumes ZFS ne peuvent pas être ajoutés à une zone non globale à l'aide de la sous-commande `add dataset` de la commande `zonecfg`. Il est cependant possible d'ajouter des volumes à une zone à l'aide de la sous-commande `add device` de la commande `zonecfg`.

Dans l'exemple suivant, un volume ZFS est ajouté à une zone non globale par un administrateur global de la zone globale :

```
# zonecfg -z zion
zion: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zion> create
zonecfg:zion> add device
zonecfg:zion:device> set match=/dev/zvol/dsk/tank/vol
zonecfg:zion:device> end
```

Cette syntaxe ajoute le volume `tank/vol` à la zone `zion`. Notez que l'ajout d'un volume brut à une zone comporte des risques de sécurité implicites, même si le volume ne correspond pas à un périphérique physique. L'administrateur risque notamment de créer des systèmes de fichiers non conformes qui généreraient des erreurs graves dans le système en cas de tentative de montage. Pour de plus amples informations sur l'ajout de périphériques à des zones et les risques de sécurité associés, reportez-vous à la section [“Explication de la propriété zoned”](#) à la page 288.

Pour savoir comment ajouter des périphériques à des zones, reportez-vous à la [Partie II, “Zones” du Guide d'administration système : Gestion des ressources des conteneurs et des zones Oracle 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 avoir délégué un jeu de données à une zone, l'administrateur de zone peut contrôler les propriétés spécifiques au jeu de données. Lorsqu'un jeu de données est délégué à une zone, tous les ancêtres s'affichent en tant que jeux de données en lecture seule, alors que le jeu de données lui-même, ainsi que tous ses descendants, est accessible en écriture. Considérez par exemple la configuration suivante :

```
global# zfs list -Ho name
tank
tank/home
```

```
tank/data
tank/data/matrix
tank/data/zion
tank/data/zion/home
```

En cas d'ajout de `tank/data/zion` à une zone, chaque jeu de données dispose des propriétés suivantes :

Jeu de données	Visible	Accessible en écriture	Propriétés immuables
tank	Oui	Non	-
tank/home	Non	-	-
tank/data	Oui	Non	-
tank/data/matrix	Non	-	-
tank/data/zion	Oui	Oui	sharenfs, zoned, quota, reservation
tank/data/zion/home	Oui	Oui	sharenfs, zoned

Notez que chaque parent de `tank/zone/zion` est visible en lecture seule, que tous les descendants sont accessibles en écriture et que les jeux de données qui ne font pas partie de la hiérarchie parent sont invisibles. L'administrateur de zone ne peut pas modifier la propriété `sharenfs` car les zones non globales ne peuvent pas faire office de serveurs ZFS. L'administrateur de zone ne peut pas modifier la propriété `zoned` car cela entraînerait un risque de sécurité, tel que décrit dans la section suivante.

Les utilisateurs privilégiés dans la zone peuvent modifier toute autre propriété paramétrable, à l'exception des propriétés `quota` et `reservation`. Ce comportement permet à un administrateur de zone globale de contrôler l'espace disque occupé par tous les jeux de données utilisés par la zone non globale.

En outre, l'administrateur de zone globale ne peut pas modifier les propriétés `sharenfs` et `mountpoint` après la délégation d'un jeu de données à une zone non globale.

Explication de la propriété `zoned`

Lors qu'un jeu de données est délégué à une zone non globale, il doit être marqué spécialement pour que certaines propriétés ne soient pas interprétées dans le contexte de la zone globale. Lorsqu'un jeu de données est délégué à une zone non globale sous le contrôle d'un administrateur de zone, son contenu n'est plus fiable. Comme dans tous les systèmes de fichiers, cela peut entraîner la présence de binaires `setuid`, de liens symboliques ou d'autres contenus douteux qui pourraient compromettre la sécurité de la zone globale. De plus, l'interprétation de la propriété `mountpoint` est impossible dans le contexte de la zone globale. Dans le cas

contraire, l'administrateur de zone pourrait affecter l'espace de noms de la zone globale. Afin de résoudre ceci, ZFS utilise la propriété `zoned` pour indiquer qu'un jeu de données a été délégué à une zone non globale à un moment donné.

La propriété `zoned` est une valeur booléenne automatiquement activée lors de la première initialisation d'une zone contenant un jeu de données ZFS. L'activation manuelle de cette propriété par un administrateur de zone n'est pas nécessaire. Si la propriété `zoned` est définie, le montage ou le partage du jeu de données est impossible dans la zone globale. Dans l'exemple suivant, le fichier `tank/zone/zion` a été délégué à une zone, alors que le fichier `tank/zone/global` ne l'a pas été :

```
# zfs list -o name,zoned,mountpoint -r tank/zone
NAME                ZONED  MOUNTPOINT
tank/zone/global    off    /tank/zone/global
tank/zone/zion      on     /tank/zone/zion
# zfs mount
tank/zone/global    /tank/zone/global
tank/zone/zion      /export/zone/zion/root/tank/zone/zion
```

Notez la différence entre la propriété `mountpoint` et le répertoire dans lequel le jeu de données `tank/zone/zion` est actuellement monté. La propriété `mountpoint` correspond à la propriété telle qu'elle est stockée dans le disque et non à l'emplacement auquel est monté le jeu de données sur le système.

Lors de la suppression d'un jeu de données d'une zone ou de la destruction d'une zone, la propriété `zoned` *n'est pas* effacée automatiquement. Ce comportement est dû aux risques de sécurité inhérents associés à ces tâches. Dans la mesure où un utilisateur qui n'est pas fiable dispose de l'accès complet au jeu de données et à ses enfants, la propriété `mountpoint` risque d'être configurée sur des valeurs erronées ou des binaires `setuid` peuvent exister dans les systèmes de fichiers.

Afin d'éviter tout risque de sécurité, l'administrateur global doit effacer manuellement la propriété `zoned` pour que le jeu de données puisse être utilisé à nouveau. Avant de configurer la propriété `zoned` sur `off`, assurez-vous que la propriété `mountpoint` du jeu de données et de tous ses enfants est configurée sur des valeurs raisonnables et qu'il n'existe aucun binaire `setuid`, ou désactivez la propriété `setuid`.

Après avoir vérifié qu'aucune vulnérabilité n'existe au niveau de la sécurité, vous pouvez désactiver la propriété `zoned` à l'aide de la commande `zfs set` ou `zfs inherit`. Si la propriété `zoned` est désactivée alors que le jeu de données est en cours d'utilisation au sein d'une zone, le système peut se comporter de façon imprévue. Ne modifiez la propriété que si vous êtes sûr que le jeu de données n'est plus en cours d'utilisation dans une zone non globale.

Utilisation de pools racine ZFS de remplacement

Lors de sa création, un pool est intrinsèquement lié au système hôte. Le système hôte gère les informations du pool. Cela lui permet de détecter l'indisponibilité de ce dernier, le cas échéant. Même si elles sont utiles dans des conditions normales d'utilisation, ces informations peuvent causer des interférences lors de l'initialisation à partir d'autres médias ou lors de la création d'un pool sur un média amovible. La fonction de pool *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

La création d'un pool racine de remplacement s'effectue le plus souvent en vue d'une utilisation avec un média amovible. Dans ces circonstances, les utilisateurs souhaitent employer un système de fichiers unique et le monter à l'emplacement de leur choix dans le système cible. Lorsqu'un pool racine de remplacement est créé à l'aide de l'option `zpool create -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 d'un autre système à l'aide de la syntaxe de *valeur de la racine secondaire* `-R`.

```
# zpool export morpheus
# zpool import morpheus
cannot mount '/': directory is not empty
# zpool export morpheus
# zpool import -R /mnt morpheus
# zfs list morpheus
NAME                USED  AVAIL  REFER  MOUNTPOINT
morpheus            32.5K 33.5G   8K    /mnt
```

Importation de pools 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. Vous pouvez également utiliser cette fonction lors du montage de médias amovibles comme décrit dans la section précédente.

Dans l'exemple suivant, un pool nommé morpheus est importé à l'aide de /mnt en tant que chemin racine de remplacement : Cet exemple part du principe que morpheus a été précédemment exporté.

```
# zpool import -R /a pool
# zpool list morpheus
NAME  SIZE  ALLOC  FREE    CAP  HEALTH  ALTRoot
pool  44.8G  78K   44.7G   0%  ONLINE  /a
# zfs list pool
NAME  USED  AVAIL  REFER  MOUNTPOINT
pool  73.5K  44.1G   21K   /a/pool
```

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, manipuler des périphériques au sein d'un pool de stockage ZFS.
- Gestion de système de fichiers ZFS – Spécifie les autorisations de création, de destruction et de modification des systèmes de fichiers ZFS.

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

Outre les rôles RBAC permettant de gérer les systèmes de fichiers ZFS, vous pouvez également vous servir de l'administration déléguée de ZFS pour effectuer des tâches d'administration ZFS distribuée. Pour plus d'informations, reportez-vous au [Chapitre 9, “Administration déléguée de ZFS”](#).

Dépannage d'Oracle Solaris ZFS et récupération de pool

Ce chapitre décrit les méthodes d'identification et de résolution des pannes de ZFS. Des informations relatives à la prévention des pannes sont également fournies.

Il contient les sections suivantes :

- “Identification des défaillances ZFS” à la page 293
- “Contrôle de l'intégrité d'un système de fichiers ZFS” à la page 295
- “Résolution de problèmes avec le système de fichiers ZFS” à la page 297
- “Réparation d'un configuration ZFS endommagée” à la page 302
- “Réparation d'un périphérique manquant” à la page 303
- “Remplacement ou réparation d'un périphérique endommagé ” à la page 305
- “Réparation de données endommagées” à la page 314
- “Réparation d'un système impossible à réinitialiser” à la page 319

Identification des défaillances ZFS

En tant que système de fichiers et gestionnaire de volumes combinés, ZFS peut rencontrer différentes pannes. Ce chapitre commence par une description des différentes pannes, puis explique comment les identifier sur un système en cours d'exécution. Il se conclut en expliquant comment résoudre les problèmes. Le système de fichiers ZFS peut rencontrer trois types d'erreurs de base :

- “Périphériques manquants dans un pool de stockage ZFS” à la page 294
- “Périphériques endommagés dans un pool de stockage ZFS” à la page 294
- “Données ZFS corrompues” à la page 294

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 en état REMOVED. En fonction du niveau de réplication des données du pool, ce retrait peut résulter ou non en une indisponibilité de la totalité du pool. Le pool reste accessible en cas de suppression d'un périphérique mis en miroir ou RAID-Z. Un pool peut renvoyer l'état FAULTED. Cela signifie qu'aucune donnée n'est accessible jusqu'à ce que le périphérique soit reconnecté selon les conditions suivantes :

- si tous les composants d'un miroir sont supprimés ;
- si plusieurs périphériques d'un périphérique RAID-Z (raidz1) sont supprimés ;
- si le périphérique de niveau supérieur est supprimé dans une configuration contenant un seul disque.

Périphériques endommagés dans un pool de stockage ZFS

Le terme " endommagé " fait référence à une grande variété d'erreurs possibles. Les exemples incluent les éléments suivants :

- 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 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 du problème exact dans un périphérique n'est pas une tâche aisée. Elle est abordée plus en détail dans une section ultérieure.

Données ZFS corrompues

La corruption de données se produit lorsqu'une ou plusieurs erreurs de périphériques (indiquant un ou plusieurs périphériques manquants ou endommagés) affectent un périphérique virtuel de niveau supérieur. Par exemple, la moitié d'un miroir peut subir des

milliers d'erreurs sans jamais causer 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 pools 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.

Contrôle de l'intégrité d'un système de fichiers ZFS

Il n'existe pas d'utilitaire `fsck` équivalent pour ZFS. Cet utilitaire remplissait deux fonctions : réparer et valider le système de fichiers.

Réparation du système de fichiers

Avec les systèmes de fichiers classiques, la méthode d'écriture des données est affectée par les pannes inattendues entraînant des incohérences de systèmes de fichiers. Un système de fichiers classique n'étant pas transactionnel, les blocs non référencés, les comptes de liens défectueux ou autres structures de systèmes de fichiers incohérentes sont possibles. L'ajout de la journalisation résout certains de ces problèmes, mais peut entraîner des problèmes supplémentaires lorsque la restauration du journal est impossible. Une incohérence des données sur disque dans une configuration ZFS ne se produit qu'à la suite d'une panne de matérielle (auquel cas le pool aurait dû être redondant) ou en présence d'un bogue dans le logiciel ZFS.

L'utilitaire `fsck` répare les problèmes connus spécifiques aux systèmes de fichiers UFS. La plupart des problèmes au niveau des pools de stockage ZFS sont généralement liés à un matériel défaillant ou à des pannes de courant. En utilisant des pools redondants, vous pouvez éviter de nombreux problèmes. Si le pool est endommagé suite à une défaillance de matériel ou à une coupure de courant, reportez-vous à la section [“Réparation de dommages présents dans l'ensemble du pool de stockage ZFS”](#) à la page 317.

Si le pool n'est pas redondant, le risque qu'une corruption de système de fichiers puisse rendre tout ou partie de vos données inaccessibles est toujours présent.

Validation du système de fichiers

Outre la réparation du système de fichiers, l'utilitaire `fsck` valide l'absence de problème relatif aux données sur le disque. Cette tâche requiert habituellement le démontage du système de fichiers et en l'exécution de l'utilitaire `fsck`, éventuellement en mettant le système en mode

utilisateur unique lors du processus. Ce scénario entraîne une indisponibilité proportionnelle à la taille du système de fichiers en cours de vérification. Plutôt que de requérir un utilitaire explicite pour effectuer la vérification nécessaire, ZFS fournit un mécanisme pour effectuer une vérification de routine des incohérences. Cette fonctionnalité, appelée *nettoyage*, est fréquemment utilisée dans les systèmes de mémoire et autres systèmes comme méthode de détection et de prévention d'erreurs pour éviter qu'elles entraînent des pannes matérielles ou logicielles.

Contrôle du nettoyage de données ZFS

Si ZFS rencontre une erreur, soit via le nettoyage ou lors de l'accès à un fichier à la demande, l'erreur est journalisée en interne pour vous donner une vue d'ensemble rapide de toutes les erreurs connues au sein du pool.

Nettoyage explicite de données ZFS

La façon la plus simple de vérifier l'intégrité des données est de lancer un nettoyage explicite de toutes les données au sein du pool. Cette opération traverse toutes les données dans le pool une fois et vérifie que tous les blocs sont lisibles. Le nettoyage va aussi vite que le permettent les périphériques, mais la priorité de toute E/S reste inférieure à celle de toute opération normale. Cette opération peut affecter les performances, bien que les données du pool restent utilisables et leur réactivité quasiment la même lors du nettoyage. La commande `zpool scrub` permet de lancer un nettoyage explicite. Exemple :

```
# zpool scrub tank
```

La commande `zpool status` ne permet pas d'afficher l'état de l'opération de nettoyage actuelle. Exemple :

```
# zpool status -v tank
pool: tank
state: ONLINE
scrub: scrub completed after 0h7m with 0 errors on Tue Tue Feb  2 12:54:00 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c1t0d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0

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

Une seule opération de nettoyage actif par pool peut se produire à la fois.

L'option `-s` permet d'interrompre une opération de nettoyage en cours. Exemple :

```
# zpool scrub -s tank
```


Dans la plupart des cas, une opération de nettoyage pour assurer l'intégrité des données doit être menée à son terme. Vous pouvez cependant interrompre une telle opération si les performances du système sont affectées.

Un nettoyage de routine garantit des E/S continues pour l'ensemble des disques du système. Cette 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 107.

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 resynchronisation suspend le nettoyage et le redémarre une fois qu'elle a terminé.

Pour de plus amples informations sur la resynchronisation, reportez-vous à la section [“Affichage de l'état de réargenture”](#) à la page 313.

Résolution de problèmes avec le système de fichiers ZFS

Les sections suivantes décrivent l'identification et la résolution des problèmes dans les systèmes de fichiers ZFS ou les pools de stockage :

- [“Recherche de problèmes éventuels dans un pool de stockage ZFS”](#) à la page 299
- [“Consultation de la sortie de `zpool status`”](#) à la page 299
- [“Rapport système de messages d'erreur ZFS”](#) à la page 302

Les fonctions suivantes permettent d'identifier les problèmes au sein de la configuration ZFS :

- La commande `zpool status` permet d'afficher les informations détaillées des pools de stockage ZFS.
- Les défaillances de pool et de périphérique sont rapportées par le biais de messages de diagnostics ZFS/FMA.
- La commande `zpool history` permet d'afficher les commandes ZFS précédentes qui ont modifié les informations d'état de pool.

La commande `zpool status` permet de résoudre la plupart des problèmes au niveau de ZFS. Cette commande analyse les différentes erreurs système et identifie les problèmes les plus sévères. En outre, elle propose des actions à effectuer et un lien vers un article de connaissances pour de plus amples informations. Notez que cette commande n'identifie qu'un seul problème

dans le pool, même si plusieurs problèmes existent. Par exemple, les erreurs de corruption de données sont généralement provoquées par la panne d'un périphérique, mais le remplacement d'un périphérique défaillant peut ne pas résoudre tous les problèmes de corruption de données.

En outre, un moteur de diagnostic ZFS diagnostique et signale les défaillances au niveau du pool et des périphériques. Les erreurs liées aux sommes de contrôle, aux E/S, aux périphériques et aux pools font également l'objet d'un rapport lorsqu'elles sont liées à ces défaillances. Les défaillances ZFS telles que rapportées par `fmd` s'affichent sur la console ainsi que dans le fichier de messages système. Dans la plupart des cas, le message `fmd` vous dirige vers la commande `zpool status` pour obtenir des instructions supplémentaires de récupération.

Le processus de récupération est comme décrit ci-après :

- Le cas échéant, la commande `zpool history` permet d'identifier les commandes ZFS ayant précédé le scénario d'erreur. Exemple :

```
# zpool history tank
History for 'tank':
2010-07-15.12:06:50 zpool create tank mirror c0t1d0 c0t2d0 c0t3d0
2010-07-15.12:06:58 zfs create tank/erick
2010-07-15.12:07:01 zfs set checksum=off tank/erick
```

Dans cette sortie, notez 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 `fmd` affichés sur la console système ou dans le fichier `/var/adm/messages`.
- Obtenez des instructions de réparation supplémentaires grâce à la commande `zpool status -x`.
- Réparez les pannes. Pour ce faire, suivez les étapes suivantes :
 - 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.

Cette section explique comment interpréter la sortie `zpool status` afin de diagnostiquer le type de défaillances pouvant survenir. Même si la commande effectue automatiquement le travail, vous devez comprendre exactement les problèmes identifiés pour diagnostiquer la panne. Les sections suivantes expliquent comment corriger les différents types de problèmes que vous risquez de rencontrer.

Recherche de problèmes éventuels dans un pool de stockage ZFS

La méthode la plus simple pour déterminer s'il existe des problèmes connus sur le système consiste à exécuter la commande `zpool status -x`. Cette commande décrit uniquement les pools présentant des problèmes. Si tous les pools du système fonctionnent correctement, la commande affiche les éléments suivants :

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

Sans l'indicateur `-x`, la commande affiche l'état complet de tous les pools (ou du pool demandé s'il est spécifié sur la ligne de commande), même si les pools sont autrement fonctionnels.

Pour de plus amples informations sur les options de ligne de commande de la commande `zpool status`, reportez-vous à la section [“Requête d'état de pool de stockage ZFS”](#) à la page 107.

Consultation de la sortie de `zpool status`

La sortie complète de `zpool status` est similaire à ce qui suit :

```
# zpool status tank
# zpool status tank
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: none requested
config:
```

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

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

Cette sortie est décrite ci-dessous :

Informations globales d'état des pools

Cette section de la sortie `zpool status` se compose des champs suivants, certains d'entre eux n'étant affichés que pour les pools présentant des problèmes :

- `pool` Identifie le nom du pool.
- `state` Indique l'état de maintenance actuel du pool. Ces informations concernent uniquement la capacité de pool à fournir le niveau de réplication requis.

<code>status</code>	Décrit les problèmes du pool. Ce champ est absent si aucune erreur n'est détectée.
<code>action</code>	Action recommandée pour la réparation des erreurs. Ce champ est absent si aucune erreur n'est détectée.
<code>see</code>	Fait référence à un article de connaissances contenant des informations de réparation détaillées. Les articles en ligne sont mis à jour plus régulièrement que ce guide. Par conséquent, vous devez vous y reporter pour obtenir les procédures de réparation les plus récentes. Ce champ est absent si aucune erreur n'est détectée.
<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 connues.

Informations de configuration de pool

Le champ `config` de la sortie `zpool status` décrit la configuration des périphériques inclus dans le pool, ainsi que leur état et toute erreur générée à partir des périphériques. L'état peut être l'un des suivants : `ONLINE`, `FAULTED`, `DEGRADED`, `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` : erreurs d'E/S qui se sont produites lors de l'envoi d'une requête de lecture
- `WRITE` : erreurs d'E/S qui se sont produites lors de l'envoi d'une requête d'écriture
- `CKSUM` : erreurs de somme de contrôle signifiant que le périphérique a renvoyé des données corrompues en réponse à une demande de lecture.

Il est possible d'utiliser ces erreurs pour déterminer si les dommages sont permanents. Des erreurs d'E/S peu nombreuses peuvent indiquer une interruption de service temporaire. Si elles sont nombreuses, il est possible que le périphérique présente un problème permanent. Ces erreurs ne correspondent pas nécessairement à la corruption de données telle qu'interprétée par les applications. Si la configuration du périphérique est redondante, les périphériques peuvent présenter des erreurs impossibles à corriger, même si aucune erreur ne s'affiche au niveau du périphérique RAID-Z ou du miroir. Dans ce cas, ZFS a récupéré les données correctes et a réussi à réparer les données endommagées à partir des répliques existantes.

Pour de plus amples informations sur l'interprétation de ces erreurs, reportez-vous à la section [“Détermination du type de panne de périphérique”](#) à la page 305.

Enfin, les informations auxiliaires supplémentaire sont affichées dans la dernière colonne de la sortie de `zpool status`. Ces informations s'étendent dans le champ `state` et facilitent le diagnostic des pannes. Si l'état d'un périphérique est `FAULTED`, ce champ indique si périphérique

est inaccessible ou si les données du périphérique sont corrompues. Si le périphérique est en cours de resynchronisation, ce champ affiche la progression du processus.

Pour de plus amples informations sur le contrôle de la progression de la resynchronisation, reportez-vous à la section [“Affichage de l'état de réargenture”](#) à la page 313.

État du nettoyage

La section sur le nettoyage de la sortie `zpool status` décrit l'état actuel de toute opération de nettoyage explicite. Ces informations sont distinctes de la détection d'erreurs dans le système, mais il est possible de les utiliser pour déterminer l'exactitude du rapport d'erreurs de corruption de données. Si le dernier nettoyage s'est récemment terminé, toute corruption de données existante aura probablement déjà été détecté.

Les messages de fin de nettoyage subsistent après plusieurs réinitialisations du système.

Pour de plus amples informations sur le nettoyage de données et l'interprétation de ces informations, reportez-vous à la section [“Contrôle de l'intégrité d'un système de fichiers ZFS”](#) à la page 295.

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 des données ou lors des opérations normales. Le système de fichiers ZFS gère un journal persistant de toutes les erreurs de données associées à un pool. Ce journal tourne à chaque fois qu'un nettoyage complet du système est terminé.

Les erreurs 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: UNAVAIL
status: One or more devices are faulted in response to IO failures.
action: Make sure the affected devices are connected, then run 'zpool clear'.
       see: http://www.sun.com/msg/ZFS-8000-HC
scrub: scrub completed after 0h0m with 0 errors on Tue Feb  2 13:08:42 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM	
tank	UNAVAIL	0	0	0	insufficient replicas
clt0d0	ONLINE	0	0	0	
clt1d0	UNAVAIL	4	1	0	cannot open

```
errors: Permanent errors have been detected in the following files:
```

```
/tank/data/aaa  
/tank/data/bbb  
/tank/data/ccc
```

La commande `fmfdump` affiche également un message similaire dans la console système et le fichier `/var/adm/messages`. La commande `fmfdump` 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 315.

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 produite 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 `fmfdump`.

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 gère un cache de pools actifs et les configurations correspondantes dans le système de fichiers racine. Si ce fichier de cache est corrompu ou n'est plus synchronisé avec les informations de configuration stockées dans le disque, l'ouverture du pool n'est plus possible. Le système de fichiers ZFS tente d'éviter cette situation, même si des corruptions arbitraires peuvent toujours survenir en raison des caractéristiques du système de fichiers sous-jacent et du

stockage. En général, cette situation est due à la disparition d'un pool du système alors qu'il devrait être disponible. Parfois, elle correspond à une configuration partielle, dans laquelle il manque un nombre inconnu de périphériques virtuels de niveau supérieur. Quel que soit le cas, la configuration peut être récupérée en exportant le pool (s'il est visible à tous) et en le réimportant.

Pour de plus amples informations sur l'importation et l'exportation de pools, reportez-vous à la section [“Migration de pools de stockage ZFS”](#) à la page 116.

Réparation d'un périphérique manquant

Si l'ouverture d'un périphérique est impossible, ce dernier s'affiche dans l'état UNAVAIL dans la sortie de `zpool status`. Cet état indique que ZFS n'a pas pu ouvrir le périphérique lors du premier accès au pool ou que le périphérique est devenu indisponible par la suite. Si le périphérique rend un périphérique de niveau supérieur indisponible, l'intégralité du pool devient inaccessible. Dans le cas contraire, la tolérance de pannes du pool risque d'être compromise. Quel que soit le cas, le périphérique doit simplement être reconnecté au système pour refonctionner normalement.

Par exemple, après une panne de périphérique, `cmd` peut afficher un message similaire au suivant :

```
SUNW-MSG-ID: ZFS-8000-FD, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Thu Jun 24 10:42:36 PDT 2010
PLATFORM: SUNW,Sun-Fire-T200, CSN: -, HOSTNAME: neo2
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: a1fb66d0-cc51-cd14-a835-961c15696fcb
DESC: The number of I/O errors associated with a ZFS device exceeded
acceptable levels. Refer to http://sun.com/msg/ZFS-8000-FD for more information.
AUTO-RESPONSE: The device has been offlined and marked as faulted. An attempt
will be made to activate a hot spare if available.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.
```

Pour afficher des informations détaillées sur le problème du périphérique et sa résolution, utilisez la commande `zpool status -x`. Exemple :

```
# zpool status -x
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: scrub completed after 0h0m with 0 errors on Tue Feb 2 13:15:20 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
------	-------	------	-------	-------

```

tank          DEGRADED    0    0    0
mirror-0     DEGRADED    0    0    0
c1t0d0      ONLINE      0    0    0
c1t1d0      UNAVAIL     0    0    0  cannot open

```

errors: No known data errors

Cette sortie indique que le périphérique `c1t1d0` manquant ne fonctionne pas. Si vous estimez que le périphérique est défectueux, remplacez-le.

Exécutez ensuite la commande `zpool online` pour mettre le périphérique remplacé en ligne. Exemple :

```
# zpool online tank c1t1d0
```

Confirmez ensuite que le pool dont le périphérique a été remplacé fonctionne correctement. Exemple :

```
# zpool status -x tank
pool 'tank' is healthy
```

Reconnexion physique d'un périphérique

La reconnexion d'un périphérique dépend du périphérique en question. S'il s'agit d'un disque connecté au réseau, la connectivité au réseau doit être restaurée. S'il s'agit d'un périphérique USB ou autre 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, le périphérique doit être traité comme un périphérique endommagé. Suivez les procédures décrites dans la section [“Remplacement ou réparation d'un périphérique endommagé”](#) à la page 305.

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 système a été réinitialisé en tant que partie de la procédure `attach`, alors ZFS rebalaye automatiquement tous les périphériques lors de la tentative d'ouverture du pool. Si le pool était endommagé et que le périphérique a été remplacé alors que le système était en cours d'exécution, vous devez indiquer à ZFS que le périphérique est dorénavant disponible et qu'il est prêt à être rouvert à l'aide de la commande `zpool online`. 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 94.

Remplacement ou réparation d'un périphérique endommagé

Cette section explique comment déterminer les types de panne de périphériques, effacer les erreurs transitoires et remplacer un périphérique.

Détermination du type de panne de périphérique

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. 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.
- **Lectures ou écritures mal dirigées** – Les bogues de microprogrammes ou les pannes de matériel peuvent entraîner un référencement incorrect de l'emplacement du disque par des lectures ou écritures de blocs entiers. Ces erreurs sont généralement transitoires, mais un grand nombre d'entre elles peut indiquer un disque défectueux.
- **Erreur d'administrateur** : les administrateurs peuvent écraser par erreur des parties du disque avec des données erronées (la copie de /dev/zero sur des parties du disque, par exemple) qui entraînent la 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, y compris les erreurs d'E/S cohérentes, les transports défectueux entraînant des corruptions aléatoires ou des pannes. Ces erreurs sont typiquement permanentes.
- **Périphérique mis hors ligne** : si un périphérique est hors ligne, il est considéré comme ayant été mis hors ligne par l'administrateur, parce qu'il était défectueux. L'administrateur qui a mis ce dispositif hors ligne peut déterminer si cette hypothèse est exacte.

Il est parfois difficile de déterminer la nature exacte de la panne du dispositif. La première étape consiste à examiner le décompte d'erreurs dans la sortie de `zpool status`. Exemple :

```
# zpool status -v tpool
pool: tpool
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: scrub completed after 0h0m with 2 errors on Tue Jul 13 11:08:37 2010
config:

NAME          STATE      READ WRITE CKSUM
tpool         ONLINE    2     0     0
  ct11d0      ONLINE    2     0     0
  ct13d0      ONLINE    0     0     0
errors: Permanent errors have been detected in the following files:

    /tpool/words
```

Les erreurs sont divisées en erreurs d'E/S et en erreurs de sommes de contrôle. Ces deux catégories peuvent indiquer le type de panne possible. Une opération typique renvoie un très petit nombre d'erreurs (quelques-unes sur une longue période). Si les erreurs sont nombreuses, un périphérique est probablement en panne ou sur le point de tomber en panne. Cependant, une erreur provoquée par un administrateur peut également entraîner un grand nombre d'erreurs. Le journal système `syslog` constitue une autre source d'informations. Si le journal présente un grand nombre de messages SCSI ou de pilote Fibre Channel, il existe probablement de graves problèmes matériels. L'absence de messages `syslog` indique que les dommages sont probablement transitoires.

L'objectif est de répondre à la question suivante :

Est-il possible qu'une autre erreur se produise dans ce périphérique ?

Les erreurs qui ne se produisent qu'une fois sont considérées *transitoires* et n'indiquent pas une panne potentielle. Les erreurs suffisamment persistantes ou sévères pour indiquer une panne matérielle potentielle sont considérées comme étant des *erreurs fatales*. Aucun logiciel automatisé actuellement disponible avec ZFS ne permet de déterminer le type d'erreur. Par conséquent, l'administrateur doit procéder manuellement. Une fois l'erreur déterminée, vous pouvez réaliser l'action adéquate. En cas d'erreurs fatales, effacez les erreurs transitoires ou remplacez le périphérique. Ces procédures de réparation sont décrites dans les sections suivantes.

Même si les erreurs de périphériques sont considérées comme étant transitoires, elles peuvent tout de même entraîner des erreurs de données impossibles à corriger au sein du pool. Ces erreurs requièrent des procédures de réparation spéciales, même si le périphérique sous-jacent est considéré comme étant fonctionnel ou réparé. Pour de plus amples informations sur la réparation d'erreurs de données, reportez-vous à la section [“Réparation de données endommagées”](#) à la page 314.

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, elles peuvent être effacées en toute sécurité pour indiquer qu'aucune erreur fatale ne s'est produite. Pour effacer les compteurs d'erreurs pour les périphériques mis en miroir ou RAID-Z, utilisez la commande `zpool clear`. Exemple :

```
# zpool clear tank c1t1d0
```

Cette syntaxe efface toutes les erreurs du périphérique et tout décompte d'erreurs de données associées au périphérique.

Pour effacer toutes les erreurs associées aux périphériques virtuels du pool et tout décompte d'erreurs de données associées au pool, 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 “Effacement des erreurs de périphérique de pool de stockage” à la page 95.

Remplacement d'un périphérique dans un pool de stockage ZFS

Si le périphérique présente ou risque de présenter une panne permanente, il doit être remplacé. Le remplacement du périphérique dépend de la configuration.

- “Détermination de la possibilité de remplacement du périphérique” à la page 307
- “Périphériques impossibles à remplacer” à la page 308
- “Remplacement d'un périphérique dans un pool de stockage ZFS” à la page 309
- “Affichage de l'état de réargenture” à la page 313

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 périphérique 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 défaillants, chaque disque peut être remplacé car des répliques fonctionnelles sont disponibles. Cependant, en cas de panne de deux disques dans un périphérique virtuel RAID-Z à quatre directions (`raid-z`), 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 saine, `c1t0d0`.

```
mirror          DEGRADED
c1t0d0         ONLINE
c1t1d0         FAULTED
```

Le disque `c1t0d0` peut également être remplacé, mais un autorétablissement des données est impossible, car il n'existe aucune réplique correcte.

Dans la configuration suivante, aucun des disques défectueux ne peut être remplacé. Les disques `ONLINE` ne peuvent pas l'être non plus, car le pool lui-même est défectueux.

```
raidz          FAULTED
c1t0d0         ONLINE
c2t0d0         FAULTED
c3t0d0         FAULTED
c4t0d0         ONLINE
```

Dans la configuration suivante, chacun des disques de niveau supérieur peut être remplacé. Cependant, les données incorrectes seront également copiées dans le nouveau disque, le cas échéant.

```
c1t0d0         ONLINE
c1t1d0         ONLINE
```

Si les deux disques sont défectueux, alors tout remplacement est impossible car le pool lui-même est 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, recréer la configuration et restaurer les données à partir d'une copie de sauvegarde.

Pour de plus amples informations sur la restauration d'un pool entier, reportez-vous à la section [“Réparation de dommages présents dans l'ensemble du pool de stockage ZFS”](#) à la page 317.

Remplacement d'un périphérique dans un pool de stockage ZFS

Après avoir déterminé qu'il est possible de remplacer un périphérique, exécutez la commande `zpool replace` pour le remplacer effectivement. Exécutez la commande suivante si vous remplacez le périphérique endommagé par un autre périphérique différent :

```
# zpool replace tank c1t1d0 c2t0d0
```

Cette commande lance la migration de données vers le nouveau périphérique, soit à partir du périphérique endommagé, soit à partir d'autres périphériques du pool s'il s'agit d'une configuration redondante. Une fois l'exécution de la commande terminée, le périphérique endommagé est séparé de la configuration. Il peut dorénavant être retiré du système. Si vous avez déjà retiré le périphérique et que vous l'avez remplacé par un autre dans le même emplacement, utilisez la forme "périphérique unique" de la commande. Exemple :

```
# zpool replace tank c1t1d0
```

Cette commande formate adéquatement un disque non formaté et resynchronise ensuite les données à partir du reste de la configuration.

Pour de plus amples informations sur la commande `zpool replace` reportez-vous à la section [“Remplacement de périphériques dans un pool de stockage”](#) à la page 96.

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 en miroir `tank` sur un système Oracle 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. Voici les principales étapes à suivre :

- Déconnectez le disque (`c1t3d0`) à remplacer. Vous ne pouvez pas annuler la configuration d'un disque utilisé.
- Utilisez la commande `cfgadm` pour identifier le disque (`c1t3d0`) dont la configuration doit être annulée et annulez-la. Dans cette configuration en miroir, le pool est endommagé et le disque est hors ligne, mais le pool reste disponible.
- Remplacez le disque (`c1t3d0`). Vérifiez que la DEL bleue Ready to Remove, indiquant que le périphérique est prêt à être retiré, est allumée avant de retirer le lecteur défaillant.
- Reconfigurez le disque (`c1t3d0`).
- Mettez le nouveau disque (`c1t3d0`) en ligne.
- Exécutez la commande `zpool replace` pour remplacer le disque (`c1t3d0`).

EXEMPLE 11-1 Remplacement d'un périphérique dans un pool de stockage ZFS (Suite)

Remarque – Si vous avez précédemment défini la propriété de pool `autoreplace` sur `on`, tout nouveau périphérique détecté au même emplacement physique qu'un périphérique appartenant précédemment au pool est automatiquement formaté et remplacé sans recourir à la commande `zpool replace`. Cette fonction n'est pas prise en charge sur tous les types de matériel.

- Si un disque défectueux est automatiquement remplacé par un disque hot spare, vous devrez peut-être déconnecter le disque hot spare une fois le disque défectueux remplacé. Par exemple, si `c2t4d0` reste actif comme disque hot spare actif une fois le disque défectueux remplacé, déconnectez-le.

```
# zpool detach tank c2t4d0
```

L'exemple suivant explique étape par étape comment remplacer un disque dans un pool de stockage ZFS.

```
# zpool offline tank c1t3d0
# cfgadm | grep c1t3d0
sata1/3::dsk/c1t3d0          disk          connected    configured    ok
# cfgadm -c unconfigure sata1/3
Unconfigure the device at: /devices/pci@0,0/pci1022,7458@2/pci1lab,1lab@1:3
This operation will suspend activity on the SATA device
Continue (yes/no)? yes
# cfgadm | grep sata1/3
sata1/3          disk          connected    unconfigured    ok
<Physically replace the failed disk c1t3d0>
# cfgadm -c configure sata1/3
# cfgadm | grep sata1/3
sata1/3::dsk/c1t3d0          disk          connected    configured    ok
# zpool online tank c1t3d0
# zpool replace tank c1t3d0
# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:17:32 2010
config:
```

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

EXEMPLE 11-1 Remplacement d'un périphérique dans un pool de stockage ZFS (Suite)

```
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 resynchronisation du nouveau disque sont en cours.

Pour remplacer un disque (`c1t3d0`) par un autre disque (`c4t3d0`), il suffit d'exécuter la commande `zpool replace`. Exemple :

```
# zpool replace tank c1t3d0 c4t3d0
# zpool status
pool: tank
state: DEGRADED
scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	DEGRADED	0	0	0
mirror-0	ONLINE	0	0	0
c0t1d0	ONLINE	0	0	0
c1t1d0	ONLINE	0	0	0
mirror-1	ONLINE	0	0	0
c0t2d0	ONLINE	0	0	0
c1t2d0	ONLINE	0	0	0
mirror-2	DEGRADED	0	0	0
c0t3d0	ONLINE	0	0	0
replacing	DEGRADED	0	0	0
c1t3d0	OFFLINE	0	0	0
c4t3d0	ONLINE	0	0	0

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

La commande `zpool status` doit parfois être exécutée plusieurs fois jusqu'à la fin du remplacement du disque.

```
# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c0t1d0	ONLINE	0	0	0

EXEMPLE 11-1 Remplacement d'un périphérique dans un pool de stockage ZFS (Suite)

```

c1t1d0    ONLINE      0     0     0
mirror-1  ONLINE      0     0     0
c0t2d0    ONLINE      0     0     0
c1t2d0    ONLINE      0     0     0
mirror-2  ONLINE      0     0     0
c0t3d0    ONLINE      0     0     0
c4t3d0    ONLINE      0     0     0

```

EXEMPLE 11-2 Remplacement d'un périphérique de journal défaillant

L'exemple suivant montre comment récupérer les données d'un périphérique de journal défaillant (c0t5d0) dans le pool de stockage (pool). Voici les principales étapes à suivre :

- Vérifiez la sortie `zpool status -x` et le message de diagnostic FMA, décrits ici : <http://www.sun.com/msg/ZFS-8000-K4>
- Remplacez physiquement le périphérique de journal défaillant.
- Mettez le nouveau périphérique de journal en ligne.
- Effacez la condition d'erreur du pool.

```

# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
       Waiting for administrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
       or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:

NAME      STATE      READ WRITE CKSUM
pool      FAULTED    0     0     0 bad intent log
mirror    ONLINE     0     0     0
  c0t1d0  ONLINE     0     0     0
  c0t4d0  ONLINE     0     0     0
logs      FAULTED    0     0     0 bad intent log
  c0t5d0  UNAVAIL    0     0     0 cannot open
<Physically replace the failed log device>
# zpool online pool c0t5d0
# zpool clear pool

# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
       Waiting for administrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
       or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:

```


EXEMPLE 11-2 Remplacement d'un périphérique de journal défaillant (Suite)

```

NAME          STATE      READ WRITE CKSUM
pool          FAULTED   0     0     0 bad intent log
  mirror-0    ONLINE    0     0     0
    c0t1d0    ONLINE    0     0     0
    c0t4d0    ONLINE    0     0     0
  logs        FAULTED   0     0     0 bad intent log
    c0t5d0    UNAVAIL   0     0     0 cannot open
<Physically replace the failed log device>
# zpool online pool c0t5d0
# zpool clear pool

```

Affichage de l'état de réargenture

Le processus de remplacement d'un périphérique peut prendre beaucoup de temps, selon la taille du périphérique et la quantité de données dans le pool. Le processus de déplacement de données d'un périphérique à un autre s'appelle la *resynchronisation*. Vous pouvez la contrôler à l'aide de la commande `zpool status`.

Les systèmes de fichiers traditionnels effectuent la resynchronisation 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 resynchronisation de façon bien plus puissante et contrôlée. Les deux avantages de cette fonction sont comme suite :

- ZFS n'effectue la resynchronisation que de la quantité minimale de données requises. Dans le cas d'une brève interruption de service (par rapport à un remplacement complet d'un périphérique), vous pouvez effectuer la resynchronisation du disque en quelques minutes ou quelques secondes. Lors du remplacement d'un disque entier, la durée du processus de resynchronisation 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: DEGRADED
status: One or more devices is currently being resilvered. The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scrub: resilver in progress for 0h0m, 22.60% done, 0h1m to go
config:
NAME          STATE      READ WRITE CKSUM
tank          DEGRADED   0     0     0
  mirror-0    DEGRADED   0     0     0

```

```

replacing-0 DEGRADED      0      0      0
  c1t0d0    UNAVAIL      0      0      0  cannot open
  c2t0d0    ONLINE       0      0      0  85.0M resilvered
c1t1d0     ONLINE       0      0      0

```

errors: No known data errors

Dans cet exemple, le disque `c1t0d0` est remplacé par `c2t0d0`. Cet événement est observé dans la sortie d'état par la présence du périphérique virtuel `replacing` de la configuration. Ce périphérique n'est pas réel et ne permet pas de créer un pool. L'objectif de ce périphérique consiste uniquement à afficher le processus de resynchronisation et à identifier le périphérique en cours de remplacement.

Notez que tout pool en cours de resynchronisation 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 resynchronisation s'effectue aussi rapidement que possible, mais les E/S sont toujours programmées avec une priorité inférieure à celle des E/S requises par l'utilisateur afin de minimiser l'impact sur le système. Une fois la resynchronisation terminée, la nouvelle configuration complète s'applique, remplaçant l'ancienne configuration. Exemple :

```

# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h1m with 0 errors on Tue Feb  2 13:54:30 2010
config:

```

```

NAME      STATE      READ WRITE CKSUM
tank      ONLINE     0     0     0
  mirror-0 ONLINE     0     0     0
    c2t0d0 ONLINE     0     0     0  377M resilvered
    c1t1d0 ONLINE     0     0     0

```

errors: No known data errors

L'état du pool est à nouveau `ONLINE` et le disque défectueux d'origine (`c1t0d0`) a été supprimé de la configuration.

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 315](#)
- [“Réparation d'un fichier ou répertoire corrompu” à la page 316](#)
- [“Réparation de dommages présents dans l'ensemble du pool de stockage ZFS” à la page 317](#)

ZFS utilise les données des sommes de contrôles, de redondance et d'auto-rétablissement pour minimiser le risque 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 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 [“Contrôle de l'intégrité d'un système de fichiers ZFS”](#) à la page 295.

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 monkey
pool: monkey
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
monkey	ONLINE	8	0	0
c1t1d0	ONLINE	2	0	0
c2t5d0	ONLINE	6	0	0

```
errors: 8 data errors, use '-v' for a list
```

Toute erreur indique seulement qu'une erreur s'est produite à un moment donné. Il est possible que certaines erreurs ne soient plus présentes dans le système. Dans le cadre d'une utilisation normale, elles le sont. Certaines interruptions de service temporaires peuvent entraîner une 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 devez 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. Utilisez la commande `zpool status -v` pour afficher la liste des noms de fichier contenant 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: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:

      NAME      STATE      READ WRITE CKSUM
      monkey    ONLINE    8     0     0
      c1t1d0    ONLINE    2     0     0
      c2t5d0    ONLINE    6     0     0
```

errors: Permanent errors have been detected in the following files:

```

/monkey/a.txt
/monkey/bananas/b.txt
/monkey/sub/dir/d.txt
monkey/ghost/e.txt
/monkey/ghost/boo/f.txt

```

La liste des noms de fichiers comportant des erreurs persistantes peut être décrite comme suit :

- Si le chemin complet du fichier est trouvé et si le jeu de données est monté, le chemin complet du fichier s'affiche. 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 les fichiers ou les répertoires vers un autre emplacement. Cela permet de restaurer l'objet d'origine à son emplacement.

Réparation de dommages présents dans l'ensemble du pool de stockage ZFS

Si les métadonnées du pool sont endommagées et que cela empêche l'ouverture ou l'importation du pool, vous pouvez utiliser les options suivantes :

- Tentez de récupérer le pool à l'aide de la commande `zpool clear -F` ou `zpool import -F`. Ces commandes tentent d'annuler (roll back) les dernières transactions restantes du pool pour qu'elles reviennent à un fonctionnement normal. Vous pouvez utiliser la commande `zpool status` pour vérifier le pool endommagé et les mesures de récupération recommandées. Exemple :

```

# zpool status
pool: tpool
state: FAULTED
status: The pool metadata is corrupted and the pool cannot be opened.
action: Recovery is possible, but will result in some data loss.

```

```
Returning the pool to its state as of Wed Jul 14 11:44:10 2010
should correct the problem. Approximately 5 seconds of data
must be discarded, irreversibly. Recovery can be attempted
by executing 'zpool clear -F tpool'. A scrub of the pool
is strongly recommended after recovery.
```

```
see: http://www.sun.com/msg/ZFS-8000-72
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM	
tpool	FAULTED	0	0	1	corrupted data
c1t1d0	ONLINE	0	0	2	
c1t3d0	ONLINE	0	0	4	

Le processus de récupération comme décrit ci-dessus consiste à utiliser la commande suivante :

```
# zpool clear -F tpool
```

Si vous tentez d'importer un pool de stockage endommagé, des messages semblables aux messages suivants s'affichent :

```
# zpool import tpool
cannot import 'tpool': I/O error
Recovery is possible, but will result in some data loss.
Returning the pool to its state as of Wed Jul 14 11:44:10 2010
should correct the problem. Approximately 5 seconds of data
must be discarded, irreversibly. Recovery can be attempted
by executing 'zpool import -F tpool'. A scrub of the pool
is strongly recommended after recovery.
```

Le processus de récupération comme décrit ci-dessus consiste à utiliser la commande suivante :

```
# zpool import -F tpool
Pool tpool returned to its state as of Wed Jul 14 11:44:10 2010.
Discarded approximately 5 seconds of transactions
```

Si le pool endommagé se trouve dans le fichier `zpool.cache`, le problème est détecté lors de l'initialisation du système. Le pool endommagé est consigné dans la commande `zpool status`. Si le pool ne se trouve pas dans le fichier `zpool.cache`, il n'est pas importé ou ouvert et des messages indiquant que le pool est endommagé s'affichent lorsque vous tentez de l'importer.

- Si le pool ne peut pas être récupéré par le biais de la méthode de récupération ci-dessus, vous devez restaurer le pool et l'ensemble de ses données à partir d'une copie de sauvegarde. Le mécanisme utilisé varie énormément selon la configuration du pool et la stratégie de sauvegarde. Tout d'abord, enregistrez la configuration telle qu'elle s'affiche dans la commande `zpool status` pour pouvoir 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=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 dans le paragraphe ci-dessus.
- Utilisez la commande `fmdump -eV` pour afficher les pools présentant des erreurs fatales et déterminer ainsi quel pool pose des problèmes.
- Importez les pools un à un en ignorant ceux qui posent problème, comme décrit dans la sortie de la commande `fmdump`.

Descriptions des versions d'Oracle Solaris ZFS

Cette annexe décrit les versions ZFS disponibles, les fonctionnalités de chacune d'entre elles et le SE Solaris correspondant.

L'annexe contient les sections suivantes :

- “Présentation des versions ZFS” à la page 321
- “Versions de pool ZFS” à la page 321
- “Versions du système de fichiers ZFS” à la page 322

Présentation des versions ZFS

Les nouvelles fonctions de pool et de système de fichiers ZFS ont été introduites dans certaines versions du système de fichiers ZFS disponibles avec Solaris. Vous pouvez utiliser la commande `zpool upgrade` ou `zfs upgrade` pour déterminer si la version d'un pool ou un système de fichiers est antérieure à la version Solaris en cours d'exécution. Vous pouvez également utiliser ces commandes pour mettre à niveau la version de votre pool et de votre système de fichiers.

Pour plus d'informations sur l'utilisation des commandes `zpool upgrade` et `zfs upgrade`, reportez-vous aux sections “Mise à niveau des systèmes de fichiers ZFS (`zfs upgrade`)” à la page 33 et “Mise à niveau de pools de stockage ZFS” à la page 123.

Versions de pool ZFS

Le tableau suivant présente une liste des versions de pool ZFS disponibles dans les versions précédentes de Solaris.

Version	Solaris10	Description
1	Solaris 10 6/06	Version ZFS initiale

Version	Solaris10	Description
2	Solaris 10 11/06	Blocs ditto (métadonnées répliquées)
3	Solaris 10 11/06	Disques hot spare et RAID-Z à double parité
4	Solaris 10 8/07	zpool history
5	Solaris 10 10/08	Algorithme de compression gzip
6	Solaris 10 10/08	Propriété de pool boot fs
7	Solaris 10 10/08	Périphérique de journal de tentatives distincts
8	Solaris 10 10/08	Administration déléguée
9	Solaris 10 10/08	Propriétés refquota et refreservation
10	Solaris 10 5/09	Périphériques de cache
11	Solaris 10 10/09	Performances de nettoyage améliorées
12	Solaris 10 10/09	Propriétés d'instantané
13	Solaris 10 10/09	Propriété snapused
14	Solaris 10 10/09	Propriété Aclinherit passthrough-x
15	Solaris 10 10/09	Comptabilisation des espaces de groupe et d'utilisateur
16	Solaris 10 9/10	Prise en charge de la propriété stmf
17	Solaris 10 9/10	RAID-Z triple parité
18	Solaris 10 9/10	Instantanés conservés par l'utilisateur
19	Solaris 10 9/10	Suppression d'un périphérique de journal
20	Solaris 10 9/10	Compression using zle (encodage de chaîne vide)
21	Solaris 10 9/10	Réservée
22	Solaris 10 9/10	Propriétés reçues

Versions du système de fichiers ZFS

Le tableau suivant répertorie les versions du système de fichiers ZFS disponibles dans les versions précédentes de Solaris.

Version	Solaris10	Description
1	Solaris 10 6/06	Version initiale du système de fichiers ZFS
2	Solaris 10 10/08	Entrées de répertoire améliorées

Version	Solaris10	Description
3	Solaris 10 10/08	Insensibilité à la casse et identificateur unique de système de fichiers (FUID)
4	Solaris 10 10/09	Propriétés userquota et groupquota

Index

A

Accès

- Instantané ZFS
(exemple), 230

ACL

- ACL sur un fichier ZFS
 - Description détaillée, 249
- ACL sur un répertoire ZFS
 - Description détaillée, 250
- Configuration d'ACL dans un fichier ZFS (format compact)
 - Exemple, 263
- Définition d'ACL sur un fichier ZFS (format compact)
 - Description, 262
- Définition d'une ACL sur un fichier ZFS (format détaillé)
 - Fichier de description, 250
- Définition de l'héritage d'une ACL dans un fichier ZFS (format détaillé)
 - Exemple, 255
- Définition sur les fichiers ZFS
 - Description, 248
- Description, 241
- Description de format, 243
- Différences avec les ACL POSIX-draft, 242
- Héritage d'ACL, 246
- Indicateurs d'héritage d'ACL, 246
- Modification d'une ACL triviale sur un fichier ZFS (format détaillé)
 - Exemple, 251
- Privilèges d'accès, 244

ACL (*Suite*)

- Propriété `aclinherit`, 247
- Propriété `aclmode`, 247
- Propriété d'ACL, 247
- Restauration d'une ACL triviale sur un fichier ZFS (format détaillé)
 - Exemple, 255
 - Type d'entrée, 244
- ACL NFSv4
 - Indicateurs d'héritage d'ACL, 246
 - Modèle
 - Description, 241
- ACL Solaris
 - Description de format, 243
 - Différences avec les ACL POSIX-draft, 242
 - héritage d'ACL, 246
 - Indicateurs d'héritage d'ACL, 246
 - Nouveau modèle
 - Description, 241
 - Propriété d'ACL, 247
- ACL NFSv4
 - Description de format, 243
 - Différences avec les ACL POSIX-draft, 242
 - Héritage d'ACL, 246
 - Propriété d'ACL, 247
- ACL POSIX-draft, Description, 242
- Administration déléguée, présentation, 269
- Administration déléguée de ZFS, présentation, 269
- Administration simplifiée, Description, 49
- Affichage
 - Droits délégués (exemple), 278

Affichage (Suite)

- État de maintenance d'un pool de stockage ZFS
 - Exemple, 114
- État détaillé du fonctionnement du pool de stockage ZFS
 - Exemple, 115
- État fonctionnel d'un pool de stockage
 - Description, 113
- history, commande, 40
- Rapport syslog de messages d'erreur ZFS
 - Description, 302
- Statistiques d'E/S à l'échelle du pool de stockage ZFS
 - Exemple, 111
- Statistiques d'E/S de pool de stockage vdev ZFS
 - Exemple, 112
- Statistiques d'E/S de pools de stockage ZFS
 - Description, 111
- Ajout
 - Disques, configuration RAID-Z (exemple), 84
 - Périphérique à un pool de stockage ZFS (zpool add)
 - Exemple, 83
 - Périphérique de journal en miroir (exemple), 85
 - Périphériques de cache (exemple), 86
 - Système de fichiers ZFS à une zone non globale
 - Exemple, 285
 - Volume ZFS à une zone non globale (exemple), 286
- Ajustement, Taille de périphériques de swap et de vidage, 168
- allocated (propriété), description, 104
- Annulation du partage
 - Système de fichiers ZFS
 - exemple, 217
- atime, Propriété, Description, 190
- autoreplace (propriété), description, 105
- Autorétablissement, Description, 71
- available (propriété), Description, 191

B

- Bloc d'initialisation, Installation à l'aide de installboot et de installgrub, 171

C

- cachefile (propriété), description, 105
- canmount (propriété)
 - Description, 191
 - Description détaillée, 201
- checksum, propriété, Description, 191
- Clone
 - Création, exemple, 232
 - Définition, 50
 - Destruction, exemple, 233
 - Fonctions, 232
- Comportement d'espace saturé, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 63
- Composant ZFS, Convention d'attribution de nom, 52
- Composants, Pool de stockage ZFS, 65
- compression, propriété, Description, 192
- compressratio (propriété), Description, 192
- Comptabilisation de l'espace ZFS, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 62
- Configuration en miroir
 - Description, 69
 - Fonction de redondance, 69
 - Vue conceptuelle, 69
- Configuration RAID-Z
 - Double parité, description, 69
 - Exemple, 74
 - Fonction de redondance, 69
 - Parité simple, description, 69
 - Vue conceptuelle, 69
- Configuration RAID-Z, ajout de disques, Exemple, 84
- Configuration requise, Installation et Oracle Solaris Live Upgrade, 127
- Connexion
 - Périphérique, à un pool de stockage ZFS (zpool attach)
 - Exemple, 88
- Contrôle
 - Intégrité des données ZFS, 295
 - Validation des données (nettoyage), 296
- Convention d'attribution de nom, Composant ZFS, 52
- copies, propriété, Description, 192

Création

- Clone ZFS, exemple, 232
- Hierarchie d'un système de fichiers ZFS, 56
- Instantané ZFS
 - (exemple), 226
- Nouveau pool par scission d'un pool de stockage mis en miroir (`zpool split`, commande)
 - Exemple, 89
- Pool de stockage, à l'aide d'un périphérique de journal (exemple), 75
- Pool de stockage avec périphérique de cache (exemple), 76
- Pool de stockage RAID-Z à double parité (`zpool create`, commande)
 - Exemple, 74
- Pool de stockage RAID-Z à parité simple (`zpool create`)
 - Exemple, 74
- Pool de stockage RAID-Z à triple parité (`zpool create`, commande)
 - Exemple, 74
- Pool de stockage ZFS
 - Description, 72
- Pool de stockage ZFS (`zpool create`)
 - Exemple, 54, 72
- Pool de stockage ZFS mis en miroir (`zpool create`) (exemple), 72
- Pool racine de remplacement
 - Exemple, 290
- Système de fichiers ZFS, 58
 - (exemple), 186
 - Description, 186
- Système de fichiers ZFS de base (`zpool create`)
 - Exemple, 54
- Volume ZFS
 - Exemple, 281
- `creation` (propriété), description, 192

D

Définition

- ACL dans un fichier ZFS (format compact)
 - Exemple, 263

Définition (*Suite*)

- ACL sur des fichiers ZFS
 - Description, 248
- ACL sur un fichier ZFS (format compact)
 - Description, 262
- ACL sur un fichier ZFS (format détaillé)
 - Description, 250
- `atime`, propriété ZFS
 - exemple, 206
- `compression`, propriété
 - Exemple, 58
- Héritage d'une ACL dans un fichier ZFS (format détaillé)
 - Exemple, 255
- `mountpoint`, propriété, 58
- Point de montage ZFS (`zfs set mountpoint`)
 - Exemple, 213
- Points de montage hérités (exemple), 213
- `quota`, propriété (exemple), 59
- Quota d'un système de fichiers ZFS (`zfs set quota`)
 - Exemple, 219
- Quota ZFS (exemple), 207
- Réservation de système de fichiers ZFS
 - Exemple, 222
- `sharenfs`, propriété
 - Exemple, 58
- Définition de propriétés ZFS
 - `primarycache` (propriété), 193
 - `secondarycache` (propriété), 195
 - `shareiscsi`, 196
 - `version`, 197
- Délégation
 - Droit (exemple), 274
 - Jeu de données à une zone non globale
 - Exemple, 286
 - Délégation de droits, `zfs allow`, 272
 - Délégation de droits à un groupe, (exemple), 274
 - Délégation de droits à un utilisateur individuel, (exemple), 274
- Démontage
 - Système de fichiers ZFS
 - Exemple, 215

Dépannage

- Détection de problèmes éventuels (`zpool status -x`), 299
- Déterminer si un périphérique peut être remplacé
 - Description, 308
- Panne ZFS, 293
- Périphérique endommagé, 294
- Périphérique manquant (`faulted`), 294
- Problème d'identification, 298
- Remplacement d'un périphérique (`zpool replace`) (exemple), 313
- Remplacement d'un périphérique manquant (exemple), 303
- Réparation d'un fichier ou répertoire endommagé
 - Description, 316
- Réparation d'un système qui ne peut être initialisé
 - Description, 319
- Réparation d'une configuration ZFS endommagée, 303
- Réparation de dommages au niveau d'un pool
 - Description, 319

Destruction

- Clone ZFS, exemple, 233
- Instantané ZFS (exemple), 227
- Pool de stockage ZFS
 - Description, 72
- Pool de stockage ZFS (`zpool destroy`) (exemple), 81
- Système de fichiers ZFS
 - Exemple, 187
- Système de fichiers ZFS comportant des systèmes dépendants (exemple), 188

Détection

- Niveaux de réplication incohérents
 - Exemple, 80
- Périphérique en cours d'utilisation
 - Exemple, 79

Détermination

- Type de panne de périphérique
 - Description, 305

Déterminer

- Remplacement d'un périphérique
 - Description, 308
- devices, propriété, Description, 192
- Différences entre le système de fichiers ZFS et les systèmes de fichiers standard, Nouveau modèle d'ACL standard, 63
- Différences entre un système de fichiers classique et ZFS, Granularité d'un système de fichiers, 61
- Différences existant entre les systèmes de fichiers traditionnels et ZFS
 - Comportement d'espace saturé, 63
 - Comptabilisation de l'espace ZFS, 62
 - Gestion d'un volume traditionnel, 63
 - Montage d'un système de fichiers ZFS, 63
- Disque, Composant de pool de stockage ZFS, 66
- Disque entier, Composant de pool de stockage ZFS, 66
- Disque hot spare
 - Création
 - Exemple, 98
 - Description
 - Exemple, 98

Données

- Corruption identifiée (`zpool status -v`)
 - Exemple, 301
- Endommagées, 295
- Nettoyage
 - Exemple, 296
- Réparation, 295
- Resynchronisation
 - Description, 297
- Validation (nettoyage), 296
- Données vérifiées par somme de contrôle,
 - Description, 48
- `dumpadm`, commande, Activation d'un périphérique de vidage, 170

E

Effacement

- Périphérique d'un pool de stockage ZFS (`zpool clear`)
 - Description, 95

- Enregistrement
 - Données d'un système de fichiers ZFS (`zfs send`)
 - Exemple, 235
 - Vidage mémoire sur incident
 - `savecore`, commande, 170
 - Entrelacement dynamique
 - Description, 71
 - Fonction de pool de stockage, 71
 - Envoi et réception
 - Données d'un système de fichiers ZFS
 - Description, 234
 - Etiquette EFI
 - Description, 66
 - Interaction avec ZFS, 66
 - `exec`, propriété, Description, 192
 - Exigences matérielles et logicielles, 53
 - Exportation
 - Pool de stockage ZFS
 - Exemple, 117
- F**
- `failmode` (propriété), description, 106
 - Fichiers, En tant que composants d'un pool de stockage ZFS, 68
 - Fonctionnalités des répliquions ZFS, Mise en miroir ou RAID-Z, 69
 - `free` (propriété), description, 106
- G**
- Gestion d'un volume traditionnel, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 63
 - Granularité d'un système de fichiers, Différences entre un système de fichiers classique et ZFS, 61
- H**
- Héritage
 - Propriété ZFS (`zfs inherit`)
 - Description, 207
- Hiérarchie d'un système de fichiers, Création, 56
 - history, commande, `zpool history`, 40
- I**
- Identification
 - Pool de stockage ZFS à importer (`zpool import -a`)
 - Exemple, 118
 - Stockage requis, 55
 - Type de corruption de données (`zpool status -v`)
 - Exemple, 315
 - Importation
 - Pool de stockage ZFS
 - (exemple), 121
 - Pool de stockage ZFS, à partir de répertoires alternatifs (`zpool import -d`)
 - (exemple), 120
 - Pool racine de remplacement
 - Exemple, 291
 - Initialisation
 - Environnement d'initialisation ZFS avec `boot -L` et `boot -Z` sur un système SPARC, 173
 - Système de fichiers racine, 171
 - Installation
 - Système de fichiers racine ZFS
 - Configuration requise, 127
 - Fonctions, 126
 - Installation initiale, 130
 - Installation JumpStart, 140
 - Installation de blocs d'initialisation
 - `installboot` et `installgroup`
 - Exemple, 171
 - Installation initiale du système de fichiers racine ZFS,
 - Exemple, 131
 - Installation JumpStart
 - Système de fichiers racine
 - Exemples de profils, 143
 - Problèmes, 143
 - Instantané
 - Accès
 - (exemple), 230
 - Comptabilisation de l'espace, 230
 - Création
 - (exemple), 226

Instantané (*Suite*)

- Définition, 51
- Destruction
 - (exemple), 227
- Fonctions, 225
- Renommer
 - Exemple, 228
- Restauration
 - Exemple, 231

J

Jeu de données

- Définition, 50
- Description, 186

Jeu de droits défini, 269

Journal d'intention ZFS (ZIL), Description, 34

L

Liste

- Descendant d'un système de fichiers ZFS
 - Exemple, 204
 - Informations sur le pool ZFS, 56
 - Pool de stockage ZFS
 - Description, 107
 - Pools de stockage ZFS
 - Exemple, 108
 - Propriété ZFS (`zfs list`)
 - (exemple), 208
 - Propriété ZFS par valeur source
 - (exemple), 210
 - Propriétés ZFS pour l'exécution de scripts
 - Exemple, 210
 - Système de fichiers ZFS
 - (exemple), 204
 - Système de fichiers ZFS (`zfs list`)
 - Exemple, 59
 - Système de fichiers ZFS sans l'en-tête
 - Exemple, 206
 - Types de systèmes de fichiers ZFS
 - Exemple, 205
- listsnapshots (propriété), description, 106

`luactivate`

- Système de fichier racine
 - (exemple), 148

`lucreate`

- Environnement d'initialisation ZFS à partir d'un
 - environnement d'initialisation ZFS
 - Exemple, 149

`lucreate`, commande

- Migration de système de fichiers racine
 - Exemple, 147

M

Migration

- Système de fichiers racine UFS vers système de
 - fichiers racine ZFS
 - Oracle Solaris Live Upgrade, 144
- Système de fichiers racine UFS vers un système de
 - fichiers racine ZFS
 - Problèmes, 145
- Migration d'un pool de stockage ZFS, Description, 116
- Miroir, Définition, 51
- Mise à niveau
 - Pool de stockage ZFS
 - Description, 123
- Mise en ligne d'un périphérique
 - Pool de stockage ZFS (`zpool online`)
 - Exemple, 94
- Mise en ligne et hors ligne de périphérique
 - Pool de stockage ZFS
 - Description, 93
- Mise hors ligne d'un périphérique (`zpool offline`)
 - Pool de stockage ZFS
 - Exemple, 93
- Mode de panne
 - Périphérique endommagé, 294
 - Périphérique manquant (`faulted`), 294
- Mode de propriétés d'ACL
 - `aclinherit`, 190
 - `aclmode`, 190
- Mode panne, Données endommagées, 295
- Modèle d'ACL Solaris, Différences entre le système de
 - fichiers ZFS et les systèmes de fichiers standard, 63

- Modification
 - ACL triviale sur un fichier ZFS (format détaillé)
 - Exemple, 251
 - Modification du nom
 - Système de fichiers ZFS
 - Exemple, 188
 - Montage
 - Système de fichiers ZFS
 - Exemple, 214
 - Montage d'un système de fichiers ZFS, Différences existant entre les systèmes de fichiers traditionnels et ZFS, 63
 - mot-clé de profil JumpStart, système de fichiers racine ZFS, 141
 - mounted (propriété), Description, 193
 - mountpoint, propriété, Description, 193
- N**
- Nettoyage
 - Exemple, 296
 - Validation des données, 296
 - Niveaux de réplication incohérents
 - Détection
 - Exemple, 80
 - Notification
 - périphérique reconnecté dans ZFS(zpool online)
 - Exemple, 304
- O**
- Oracle Solaris Live Upgrade
 - Migration d'un système de fichiers racine, 144
 - Migration de système de fichiers racine
 - Exemple, 147
 - Problèmes de migration d'un système de fichiers racine, 145
 - origin (propriété), Description, 193
- P**
- Panne, 293
- Partage
 - Système de fichiers ZFS
 - Description, 216
 - Exemple, 216
 - Périphérique de cache
 - Considérations d'utilisation, 76
 - Création d'un pool (exemple), 76
 - Périphérique de journal distinct, considérations, 34
 - Périphérique de journal en miroir, ajout, Exemple, 85
 - Périphérique de journalisation mis en miroir, Création d'un pool de stockage ZFS (exemple), 75
 - Périphérique de swap et de vidage
 - Ajustement de la taille, 168
 - Problèmes, 167
 - Périphérique en cours d'utilisation
 - Détection
 - Exemple, 79
 - Périphérique virtuel
 - Composant de pools de stockage ZFS, 77
 - Définition, 52
 - Périphériques de cache , suppression, (exemple), 86
 - Périphériques de cache, ajout, (exemple), 86
 - Périphériques de swap et de vidage, Description, 167
 - Point de montage
 - Automatique, 212
 - Héritage, 212
 - Par défaut pour un système de fichiers ZFS, 186
 - Valeur par défaut des pools de stockage ZFS, 81
 - Points de montage
 - Gestion dans ZFS
 - Description, 212
 - Pool, Définition, 51
 - Pool de stockage mis en miroir (zpool create), (exemple), 72
 - Pool de stockage ZFS
 - Affichage de l'état de maintenance
 - Exemple, 114
 - Affichage de l'état détaillé du fonctionnement
 - Exemple, 115
 - Affichage de l'état fonctionnel, 113
 - Affichage du processus de resynchronisation (exemple), 313
 - Ajout de périphérique (zpool add)
 - Exemple, 83

Pool de stockage ZFS (*Suite*)

- Composants, 65
- Configuration en miroir, description, 69
- Configuration RAID-Z, création (`zpool create`)
 - Exemple, 74
- Configuration RAID-Z, description, 69
- Connexion de périphériques (`zpool attach`)
 - Exemple, 88
- Corruption de données identifiée (`zpool status -v`)
 - Exemple, 301
- Création (`zpool create`)
 - Exemple, 72
- Création d'une configuration mise en miroir (`zpool create`)
 - (exemple), 72
- Destruction (`zpool destroy`)
 - (exemple), 81
- Détection de problèmes éventuels (`zpool status -x`)
 - Description, 299
- Détermination du type de panne de périphérique
 - Description, 305
- Déterminer si un périphérique peut être remplacé
 - Description, 308
- Données endommagées
 - Description, 295
- Entrelacement dynamique, 71
- Exportation
 - Exemple, 117
- Identification du type de corruption de données (`zpool status -v`)
 - Exemple, 315
- Identification pour l'importation (`zpool import -a`)
 - Exemple, 118
- Importation
 - (exemple), 121
- Importation à partir de répertoires alternatifs (`zpool import -d`)
 - (exemple), 120
- Informations globales d'état des pools pour la résolution de problèmes
 - Description, 299
- Liste
 - Exemple, 108

Pool de stockage ZFS (*Suite*)

- Message d'erreur système
 - Description, 302
- Migration
 - Description, 116
- Miroir
 - Définition, 51
- Mise à niveau
 - Description, 123
- Mise en ligne et hors ligne de périphérique
 - Description, 93
- Mise hors ligne d'un périphérique (`zpool offline`)
 - Exemple, 93
- Nettoyage de données
 - Exemple, 296
- Nettoyage des données
 - Description, 296
- Nettoyage des données et resynchronisation
 - Description, 297
- Notification d'un périphérique reconnecté dans ZFS (`zpool online`)
 - Exemple, 304
- Panne, 293
- Périphérique endommagé
 - Description, 294
- Périphérique manquant (faulted)
 - Description, 294
- Périphérique virtuel, 77
- Point de montage par défaut, 81
- Pool
 - Définition, 51
- Pool racine de remplacement, 290
- Problème d'identification
 - Description, 298
- Profil de droits, 291
- RAID-Z
 - Définition, 51
- Réargenture
 - Définition, 51
- Récupération d'un pool détruit
 - Exemple, 122
- Remplacement d'un périphérique (`zpool replace`)
 - (exemple), 96

- Pool de stockage ZFS (*Suite*)
 - Remplacement d'un périphérique manquant (exemple), 303
 - Remplacement de périphérique (zpool replace) Exemple, 309
 - Réparation d'un fichier ou répertoire endommagé Description, 316
 - Réparation d'un système qui ne peut être initialisé Description, 319
 - Réparation d'une configuration ZFS endommagée, 303
 - Réparation de dommages au niveau d'un pool Description, 319
 - Réparation des données Description, 295
 - Scission d'un pool de stockage mis en miroir (zpool split, commande) Exemple, 89
 - Script de sortie de pool de stockage Exemple, 109
 - Séparation des périphériques (zpool detach) Exemple, 89
 - Statistiques d'E/S à l'échelle du pool Exemple, 111
 - Statistiques d'E/S vdev Exemple, 112
 - Suppression d'un périphérique Exemple, 95
 - Suppression des erreurs de périphérique (zpool clear) Exemple, 307
 - Test (zpool create -n) Exemple, 80
 - Utilisation de disques entiers, 66
 - Utilisation de fichiers, 68
 - Validation des données Description, 296
 - Version Description, 321
- Pool de stockage ZFS (zpool online)
 - Mise en ligne d'un périphérique Exemple, 94
- Pool racine de remplacement
 - Création Exemple, 290
 - Description, 290
 - Importation Exemple, 291
 - Pool ZFS, propriété, version, 107
 - Propriété altroot (description), 105
 - Propriété de pool ZFS, altroot, 105
 - primarycache (propriété), Description, 193
 - Profil de droits, Gestion de systèmes de fichiers et pools de stockage ZFS, 291
 - Propriété aclinherit, 247
 - Propriété aclmode, 247
 - Propriété bootfs (description), 105
 - Propriété capacity, description, 105
 - Propriété de pool ZFS
 - allocated (propriété), 104
 - autoreplace (propriété), 105
 - bootfs, 105
 - capacity, 105
 - delegation, 105
 - failmode (propriété), 106
 - guid, 106
 - health, 106
 - listsnapshots (propriété), 106
 - size, 106
 - Propriété de ZFS
 - Description, 189
 - propriété delegation, désactivation, 270
 - Propriété delegation, description, 105
 - Propriété guid, description, 106
 - Propriété health (description), 106
 - Propriété size, description, 106
 - Propriété ZFS
 - aclinherit, 190
 - aclmode, 190
 - atime, 190
 - available, 191
 - canmount, 191
 - Description détaillée, 201
 - checksum, 191
 - compression, 192
 - compressratio, 192

Propriété ZFS (*Suite*)

- copies, 192
 - creation, 192
 - devices, 192
 - exec, 192
 - Lecture seule, 198
 - mounted, 193
 - mountpoint, 193
 - origin, 193
 - Propriété définie par l'utilisateur
 - Description détaillée, 202
 - quota, 194
 - read-only, 194
 - recordsize, 194
 - Description détaillée, 201
 - referenced, 194
 - refquota, 194
 - refreservation, 195
 - reservation, 195
 - setuid, 195
 - shareiscsi, 196
 - sharenfs, 196
 - snappdir, 196
 - type, 196
 - used, 196
 - Description détaillée, 199
 - version, 197
 - volblocksize, 197
 - volsize, 197
 - Description détaillée, 202
 - xattr, 198
 - zoned, 198
 - zoned, propriété
 - Description détaillée, 289
- Propriété ZFS définie par l'utilisateur
- Description détaillée, 202
 - Exemple, 202
- Propriété ZFS définissable, xattr, 198
- Propriété ZFS en lecture seule
- available, 191
 - compression, 192
 - creation, 192
 - Description, 198
 - mounted, 193

Propriété ZFS en lecture seule (*Suite*)

- origin, 193
 - referenced, 194
 - type, 196
 - used, 196
- Propriété ZFS pouvant être définie
- aclinherit, 190
 - aclmode, 190
 - atime, 190
 - canmount, 191
 - Description détaillée, 201
 - checksum, 191
 - compression, 192
 - copies, 192
 - exec, 192
 - mountpoint, 193
 - quota, 194
 - read-only, 194
 - recordsize, 194
 - Description détaillée, 201
 - refquota, 194
 - refreservation, 195
 - reservation, 195
 - setuid, 195
 - sharenfs, 196
 - snappdir, 196
 - used
 - Description détaillée, 199
 - volblocksize, 197
 - volsize, 197
 - Description détaillée, 202
 - zoned, 198
- Propriétés de pool ZFS
- cachefile (propriété), 105
 - free (propriété), 106
- Propriétés ZFS
- Description des propriétés héritées, 189
 - Gestion au sein d'une zone
 - Description, 287
 - Héritées, description, 189
 - Pouvant être définies, 199
 - secondarycache (propriété), 193, 195
 - usedbychildren, 196
 - usedbydataset, 197

Propriétés ZFS (*Suite*)
 usedbyreservation, 197
 usedbysnapshots, 197
 Propriétés ZFS définissables, devices, propriété, 192
 Propriétés ZFS en lecture seule
 usedbychildren, 196
 usedbydataset, 197
 usedbyreservation, 197
 usedbysnapshots, 197
 Propriétés ZFS pouvant être définies, Description, 199

Q

quota, propriété, Description, 194
 Quota et réservation, Description, 218

R

RAID-Z, Définition, 51
 read-only (propriété), Description, 194
 Réargenture, Définition, 51
 Réception
 Données de système de fichiers ZFS (zfs receive)
 (exemple), 236
 recordsize (propriété)
 Description, 194
 Description détaillée, 201
 Récupération
 Pool de stockage ZFS détruit
 Exemple, 122
 referenced (propriété), Description, 194
 refquota (propriété), Description, 194
 refreservation (propriété), Description, 195
 Remplacement
 Périphérique (zpool replace)
 (exemple), 96, 313
 Exemple, 309
 Périphérique manquant
 (exemple), 303
 Renommer
 Instantané ZFS
 Exemple, 228

Réparation
 Configuration ZFS endommagée
 Description, 303
 Dommages au niveau d'un pool
 Description, 319
 Réparation d'un fichier ou répertoire endommagé
 Description, 316
 Système qui ne peut être initialisé
 Description, 319
 reservation (propriété), description, 195
 Résolution de problèmes
 Corruption de données identifiée (zpool status -v)
 Exemple, 301
 Détermination du type de corruption de données
 (zpool status -v)
 Exemple, 315
 Détermination du type de panne de périphérique
 Description, 305
 Informations globales d'état des pools
 Description, 299
 Notification d'un périphérique reconnecté dans ZFS
 (zpool online)
 Exemple, 304
 Rapport syslog de messages d'erreur ZFS, 302
 Remplacement de périphérique (zpool replace)
 Exemple, 309
 Suppression des erreurs de périphérique (zpool
 clear)
 Exemple, 307
 Restauration
 ACL triviale sur un fichier ZFS (format détaillé)
 Exemple, 255
 Instantané ZFS
 Exemple, 231
 Resynchronisation et nettoyage des données,
 Description, 297

S

savecore, commande, Enregistrement de vidage
 mémoire sur incident, 170
 Scission d'un pool de stockage mis en miroir
 zpool split, commande
 Exemple, 89

- script
 - Sortie de pool de stockage ZFS
 - Exemple, 109
- secondarycache (propriété), Description, 195
- Sémantique transactionnelle, Description, 48
- Séparation
 - Périphérique, d'un pool de stockage ZFS (zpool detach)
 - Exemple, 89
- setuid (propriété), description, 195
- shareiscsi (propriété), Description, 196
- sharenfs, propriété
 - Description, 196, 216
- snapdir (propriété), Description, 196
- Somme de contrôle, Définition, 50
- Stockage requis, Identification, 55
- Stockage sur pool, Description, 47
- Suppression
 - Erreurs de périphérique (zpool clear)
 - Exemple, 307
 - Périphériques de cache (exemple), 86
- Suppression d'un droit, zfs unallow, 273
- Suppression d'un périphérique
 - Pool de stockage ZFS
 - Exemple, 95
- Système de fichiers, Définition, 50
- Système de fichiers racine ZFS, Problèmes de migration
 - d'un système de fichiers racine, 145
- Système de fichiers ZFS
 - ACL sur un fichier ZFS
 - Description détaillée, 249
 - ACL sur un répertoire ZFS
 - Description détaillée, 250
 - Administration simplifiée
 - Description, 49
 - Ajout d'un système de fichiers ZFS à une zone non globale
 - Exemple, 285
 - Ajout d'un volume ZFS à une zone non globale (exemple), 286
 - Annulation du partage
 - exemple, 217
 - Clone, 233
 - Définition, 50
 - Système de fichiers ZFS, Clone (*Suite*)
 - Remplacement d'un système de fichiers (exemple), 233
 - Clones
 - Description, 232
 - Comptabilisation de l'espace d'un instantané, 230
 - Configuration d'ACL dans un fichier ZFS (format compact)
 - Exemple, 263
 - Configuration requise pour l'installation et Oracle Solaris Live Upgrade, 127
 - Convention d'attribution de nom de composant, 52
 - Création
 - (exemple), 186
 - Création d'un volume ZFS
 - Exemple, 281
 - Création de clone, 232
 - Définition d'ACL sur des fichiers ZFS
 - Description, 248
 - Définition d'ACL sur un fichier ZFS (format compact)
 - Description, 262
 - Définition d'un point de montage (zfs set mountpoint)
 - Exemple, 213
 - Définition d'un point de montage hérité (exemple), 213
 - Définition d'une ACL sur un fichier ZFS (format détaillé)
 - Description, 250
 - Définition d'une réservation
 - Exemple, 222
 - Définition de l'héritage d'une ACL dans un fichier ZFS (format détaillé)
 - Exemple, 255
 - Définition de la propriété atime
 - Exemple, 206
 - Définition de la propriété quota (exemple), 207
 - Délégation d'un jeu de données à une zone non globale
 - Exemple, 286
 - Démontage
 - Exemple, 215

Système de fichiers ZFS (*Suite*)

- Description, 47, 185
- Destruction
 - Exemple, 187
- Destruction avec les systèmes dépendants (exemple), 188
- Données vérifiées par somme de contrôle
 - Description, 48
- Enregistrement d'un flux de données (zfs send)
 - Exemple, 235
- Envoi et réception
 - Description, 234
- Gestion de points de montage automatiques, 212
- Gestion de propriété au sein d'une zone
 - Description, 287
- Gestion des points de montage
 - Description, 212
- Gestion des points de montage hérités
 - description, 212
- Héritage d'une propriété (zfs inherit) (exemple), 207
- système de fichiers ZFS
 - Initialisation d'un environnement d'initialisation
 - ZFS avec boot -L et boot -Z
 - Exemple SPARC, 173
- Système de fichiers ZFS
 - Initialisation d'un système de fichiers racine
 - Description, 171
 - Installation d'un système de fichiers racine, 126
 - Installation initiale d'un système de fichiers racine ZFS, 130
 - Installation JumpStart du système de fichiers racine, 140
 - Instantané
 - Accès, 230
 - Création, 226
 - Définition, 51
 - Description, 225
 - Destruction, 227
 - Renommer, 228
 - Restauration, 231
 - Jeu de données
 - Définition, 50

Système de fichiers ZFS (*Suite*)

- Liste
 - (exemple), 204
- Liste de descendants
 - Exemple, 204
- Liste de propriétés (zfs list) (exemple), 208
- Liste de propriétés par valeur source (exemple), 210
- Liste des propriétés pour l'exécution de scripts
 - Exemple, 210
- Liste des types
 - Exemple, 205
- Liste sans en-tête
 - Exemple, 206
- Migration d'un système de fichiers racine avec Oracle Solaris Live Upgrade, 144
- Migration de système de fichier racine avec Oracle Solaris Live Upgrade
 - Exemple, 147
- Modification d'une ACL triviale sur un fichier ZFS (format détaillé)
 - Exemple, 251
- Modification du nom
 - Exemple, 188
- Montage
 - Exemple, 214
- Partage
 - Description, 216
 - Exemple, 216
- Périphérique de swap et de vidage
 - Ajustement de la taille, 168
 - Problème, 167
- Périphériques de swap et de vidage
 - Description, 167
- Point de montage par défaut
 - Exemple, 186
- Profil de droits, 291
- Réception de flux de données (zfs receive) (exemple), 236
- Restauration d'une ACL triviale sur un fichier ZFS (format détaillé)
 - exemple, 255

- Système de fichiers ZFS (*Suite*)
- Sémantique transactionnelle
 - Description, 48
 - Somme de contrôle
 - Définition, 50
 - Stockage sur pool
 - Description, 47
 - Système de fichiers
 - Définition, 50
 - Type de jeu de données
 - Description, 205
 - Utilisation d'un système Solaris avec zones installées
 - Description, 284
- système de fichiers ZFS
- Version
 - Description, 321
- Système de fichiers ZFS
- Volume
 - Définition, 52
- Système de fichiers ZFS (*zfs set quota*)
- Définition d'un quota
 - Exemple, 219
- Système de stockage ZFS
- Périphérique virtuel
 - Définition, 52
- T**
- Terminologie
- Clone, 50
 - Instantané, 51
 - Jeu de données, 50
 - Miroir, 51
 - Périphérique virtuel, 52
 - Pool, 51
 - RAID-Z, 51
 - Réargenture, 51
 - Somme de contrôle, 50
 - Système de fichiers, 50
 - Volume, 52
- Test
- Création de pool de stockage ZFS (*zpool create -n*)
 - Exemple, 80
- type, propriété, Description, 196
- Type de jeu de données, Description, 205
- U**
- used (propriété)
- Description, 196
 - Description détaillée, 199
- usedbychildren Propriété, Description, 196
- usedbydataset Propriété, Description, 197
- usedbyreservation Propriété, Description, 197
- usedbysnapshots Propriété, Description, 197
- V**
- version (propriété), Description, 197
- version propriété, description, 107
- Version ZFS
- Fonction ZFS et SE Solaris
 - Description, 321
- Vidage mémoire sur incident, Enregistrement, 170
- volblocksize (propriété), description, 197
- volsize, propriété, Description, 197
- volsize (propriété), Description détaillée, 202
- Volume, Définition, 52
- Volume ZFS, Description, 281
- X**
- xattr, Description, 198
- Z**
- zfs allow
- Affichage des droits délégués, 278
 - Description, 272
- zfs create
- (exemple), 58, 186
 - Description, 186
- zfs destroy, Exemple, 187
- zfs destroy -r, (exemple), 188
- zfs get, (exemple), 208

- zfs get -H -o, commande, Exemple, 210
- zfs get -s, (exemple), 210
- zfs inherit, Exemple, 207
- zfs list
 - (exemple), 204
 - Exemple, 59
- zfs list -H, Exemple, 206
- zfs list -r, Exemple, 204
- zfs list -t, Exemple, 205
- zfs mount, Exemple, 214
- zfs promote, Promotion d'un clone (exemple), 233
- zfs receive, (exemple), 236
- zfs rename, Exemple, 188
- zfs send, Exemple, 235
- zfs set atime, commande, Exemple, 206
- zfs set compression, Exemple, 58
- zfs set mountpoint
 - Exemple, 58, 213
- zfs set mountpoint=legacy, (exemple), 213
- zfs set quota, (exemple), 207
- zfs set quota
 - Exemple, 59, 219
- zfs set reservation, Exemple, 222
- zfs set sharenfs, Exemple, 58
- zfs set sharenfs=on, Exemple, 216
- zfs unallow, Description, 273
- zfs unmount, Exemple, 215
- Zone
 - Ajout d'un système de fichiers ZFS à une zone non globale
 - Exemple, 285
 - Délégation d'un jeu de données à une zone non globale
 - Exemple, 286
 - Gestion de propriétés ZFS au sein d'une zone
 - Description, 287
 - Utilisation de systèmes de fichiers ZFS
 - Description, 284
- zoned, propriété
 - Description, 198
 - Description détaillée, 289
- Zones
 - Ajout d'un volume ZFS à une zone non globale (exemple), 286
- Zones (*Suite*)
 - zoned, propriété
 - Description détaillée, 289
 - zpool add, Exemple, 83
 - zpool attach, Exemple, 88
 - zpool clear
 - Description, 95
 - Exemple, 95
 - zpool create
 - Exemple, 54, 56
 - Pool de base
 - Exemple, 72
 - Pool de stockage mis en miroir (exemple), 72
 - Pool de stockage RAID-Z
 - Exemple, 74
 - zpool create -n, Test (exemple), 80
 - zpool destroy, (exemple), 81
 - zpool detach, Exemple, 89
 - zpool export, Exemple, 117
 - zpool history, Exemple, 40
 - zpool import -a, Exemple, 118
 - zpool import -D, Exemple, 122
 - zpool import -d, (exemple), 120
 - zpool import *nom*, (exemple), 121
 - zpool iostat, pool complet, exemple, 111
 - zpool iostat -v, vdev, exemple, 112
 - zpool list
 - (exemple), 56
 - Description, 107
 - Exemple, 108
 - zpool list -Ho name, Exemple, 109
 - zpool offline, Exemple, 93
 - zpool online, Exemple, 94
 - zpool replace, (exemple), 96
 - zpool split, commande, Exemple, 89
 - zpool status -v, Exemple, 115
 - zpool status -x, commande, Exemple, 114
 - zpool upgrade, 123

