Gestion des systèmes de fichiers ZFS dans Oracle® Solaris 11.2

Référence: E53914-02
Décembre 2014

ORACLE

Table des matières

Utilisation de la présente documentation

- **Présentation** – décrit comment configurer et administrer les systèmes de fichiers ZFS.
- **Public visé** – administrateurs système.
- **Connaissances requises** – expérience de base en matière d'administration de systèmes UNIX ou Oracle Solaris et expérience générale en matière d'administration de systèmes de fichiers

Bibliothèque de documentation du produit

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse http://www.oracle.com/pls/topic/lookup?ctx=solaris11.

Accès aux services de support Oracle

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

Commentaires

Faites part de vos commentaires sur cette documentation à l'adresse : http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPITRE 1

Système de fichiers Oracle Solaris ZFS (introduction)

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

Ce chapitre contient les sections suivantes :

Nouveautés de ZFS pour Oracle Solaris

Les fonctionnalités ZFS suivantes ont été ajoutées à la version Oracle Solaris actuelle : dans le système de fichiers ZFS.

- Les noms à l'aide du pool temporaire

  Un stockage partagé ou de récupération dans un cas de figure, vous pouvez créer ou importer un pool avec un pool temporaire destiné nom. Pour plus d'informations, reportez-vous à la section “Importation d'un pool avec un nom temporaire” à la page 280.
- Vous surveillez la transmission de la progression d'un flux de données ZFS en temps réel. Pour plus d'informations, reportez-vous à la section “Surveillance de la progression des flux envoyés par ZFS” à la page 219.
- L'archivage, avec prise en charge de la configuration d'un unifiée configuration de la récupération de pool root est simplifié. Pour plus d'informations, reportez-vous à la section “ Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ”

Description d'Oracle Solaris ZFS

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

Stockage ZFS mis en pool

ZFS utilise le concept de *pools de stockage* pour la gestion du stockage physique. Auparavant, l'élaboration des systèmes de fichiers reposait sur un périphérique physique unique. Afin de traiter plusieurs périphériques et d'assurer la redondance de données, le concept de *gestionnaire de volume* a été introduit pour fournir la représentation d'un périphérique. Ainsi, il n'est plus nécessaire de modifier les systèmes de fichiers pour bénéficier de plusieurs périphériques. Cette conception empêchait finalement les avancées de certains systèmes de fichiers, car le système de fichiers ne pouvait pas contrôler l'emplacement 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, et la commande `fsck` ne garantissait pas la résolution de tous les problèmes. Plus récemment, les systèmes de fichiers ont introduit le concept de *journalisation*. Le processus de journalisation enregistre les actions dans un journal séparé, lequel peut ensuite être *lu* en toute sécurité en cas de panne du système. Ce processus requiert un temps système inutile car les données doivent être écrites deux fois. en outre, il entraîne souvent d'autres problèmes, par exemple l'impossibilité de relire correctement le journal.

Avec un système de fichiers transactionnel, la gestion de données s'effectue avec une sémantique de *copie lors de l'écriture*. Les données ne sont jamais écrasées et toute séquence d'opération est entièrement validée ou entièrement ignorée. L'endommagement 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.

Evolutitivé inégalée

L'évolutivité de ZFS représente l'un des éléments clés de sa conception. La taille du système de fichiers lui-même est de 128 bits et vous pouvez utiliser jusqu'à 256 quadrillions de zettaoctets de stockage. L'ensemble des métadonnées est alloué de façon dynamique. Il est donc inutile de

pré-allouer des inodes ou de limiter l'évolutivité du système de fichiers lors de sa création. Tous les algorithmes ont été écrits selon cette exigence d'évolutivité. Les répertoires peuvent contenir jusqu'à 2^{48} (256 trillions) d'entrées et le nombre de systèmes de fichiers ou de fichiers contenus dans un système de fichiers est illimité.

Instantanés ZFS

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

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

Administration simplifiée

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

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
: Un environnement d'initialisation est un environnement Oracle Solaris amorçable se composant d'un système de fichiers root ZFS et, en option,

d'autres systèmes de fichiers montés sous celui-ci. Il ne peut y avoir plus d'un environnement d'initialisation actif à la fois.

somme de contrôle
: Un hachage 256 bit de données dans un bloc de système de fichiers. La fonctionnalité de contrôle de somme regroupe entre autres, le contrôle de somme simple et rapide fletcher4 (paramètre par défaut), ainsi que les fonctions de hachage cryptographique telles que SHA256.

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

  Pour plus d'informations sur les clones, reportez-vous à la section “Présentation des clones ZFS” à la page 210.

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

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

  pool/*path*[@*snapshot*]

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

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

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

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

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

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

miroir
: A périphérique virtuel stockant des copies identiques de données sur un ou plusieurs disques. Lorsqu'un disque d'un miroir est défaillant, tout autre disque du miroir est en mesure de fournir les mêmes données.

pool
: Groupe logique de périphériques décrivant la disposition et les caractéristiques physiques du stockage disponible. L'espace disque pour les jeux de données est alloué à partir d'un pool.

	Pour plus d'informations sur les pools de stockage, reportez-vous au Chapitre 3, Gestion des pools de stockage Oracle Solaris ZFS.
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 39.
réargenture	Processus de copie de données d'un périphérique à un autre, connu sous le nom de *réargenture*. Par exemple, si un périphérique de miroir est remplacé ou mis hors ligne, les données du périphérique de miroir le plus actuel sont copiées dans le périphérique de miroir nouvellement restauré. Dans les produits de gestion de volumes classiques, ce processus est appelé *réargenture de miroir*. Pour plus d'informations sur la réargenture ZFS, reportez-vous à la section “Affichage de l'état de réargenture” à la page 305.
instantané	Copie ponctuelle en lecture seule d'un système de fichiers ou d'un volume. Pour plus d'informations sur les instantanés, reportez-vous à “Présentation des instantanés ZFS” à la page 203.
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 “Affichage des informations d'un périphérique virtuel de pool de stockage” à la page 48.
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 volumes ZFS, reportez-vous à “Volumes ZFS” à la page 269.

Exigences d'attribution de noms de composants ZFS

Chaque composant, par exemple ZFS jeux de données et les pools, doit être nommé en fonction des règles suivantes :

- Chaque composant ne peut contenir que des caractères alphanumériques en plus des quatre caractères spéciaux suivants :
  - Trait de soulignement (_)
  - Tiret (-)

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

De plus, les composants vides ne sont pas autorisés.

Différences entre les systèmes de fichiers Oracle Solaris ZFS et classiques

Granularité du système de fichiers ZFS

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 29.

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 186. 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 189. Ce modèle est très similaire au modèle dans lequel plusieurs répertoires NFS 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.

Sur un système de fichiers UFS, la commande `du` indique la taille des blocs de données au sein du fichier. Sur un système de fichiers ZFS, la commande `du` indique la taille réelle du

fichier, telle qu'elle est stockée sur le disque. La taille prend en compte les métadonnées et la compression. Ces informations vous aident à déterminer l'espace supplémentaire dont vous disposerez si vous supprimez un fichier donné. Par conséquent, même lorsque la compression est désactivée, vous obtenez des résultats différents entre ZFS et UFS.

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

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

- Pour les fichiers de volume supérieur à `recordsize`, le dernier bloc du fichier est généralement à moitié plein. Lorsque `recordsize` est défini par défaut sur 128 KO, environ 64 KO sont perdus par fichier, ce qui peut avoir un impact considérable. L'intégration de RFE 6812608 permet de remédier à ce problème. Une solution de contournement consiste à activer la compression. Même si vos données sont déjà compressées, la partie non utilisée du dernier bloc sera remplie de zéros et sera compressée sans difficulté.
- Sur un pool RAIDZ-2, chaque bloc consomme au moins 2 secteurs (par blocs de 512 octets) d'informations de parité. L'espace utilisé par les informations de parité n'est pas signalé ; toutefois, il peut varier et représenter un pourcentage beaucoup plus élevé pour les blocs de petite taille, si bien qu'il peut avoir une incidence sur les valeurs d'espace renvoyées. L'impact est plus important lorsque `recordsize` est défini sur 512 octets, où chaque bloc logique de 512 octets consomme 1,5 Ko (3 fois l'espace). Quelles que soient les données stockées, si une utilisation efficace de l'espace est primordiale, il est recommandé de conserver la valeur par défaut de `recordsize` (128 KB) et d'activer la compression (sur la valeur par défaut `lzjb`).
- La commande `df` n'a pas connaissance des données de fichiers dédupliquées.

Comportement d'espace saturé

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

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

d'espace disque. Les blocs utilisés par le fichier continuent à être référencés à partir de l'instantané.

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

Montage de système de fichiers ZFS

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

Pour plus d'informations sur le montage et le partage des systèmes de fichiers ZFS, reportez-vous à la section “Montage de système de fichiers ZFS” à la page 168.

Gestion de volumes classique

Comme décrit à la section “Stockage ZFS mis en pool” à la page 14, 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 être évitée.

Modèle ACL Solaris basé sur NFSv4

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

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

- Le modèle est basé sur la spécification NFSv4 et similaire aux ACL de type Windows NT.
- Ce modèle fournit un jeu d'autorisations d'accès plus détaillé.

- Les ACL sont définies et affichées avec les commandes `chmod` et `ls` plutôt qu'avec les commandes `setfacl` et `getfacl` .
- Une sémantique d'héritage plus riche désigne la manière dont les privilèges d'accès sont appliqués d'un répertoire à un sous-répertoire, et ainsi de suite.

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

♦ ♦ ♦ CHAPITRE 2

Mise en route d'Oracle Solaris ZFS

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

Ce chapitre contient les sections suivantes :

Profils de droits ZFS

Si vous souhaitez effectuer des tâches de gestion ZFS sans utiliser le compte superutilisateur (root), 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 plus d'informations sur la création ou l'assignation de rôles, reportez-vous à la section “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

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

Exigences et recommandations en matière de matériel et de logiciel ZFS

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

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

Création d'un système de fichiers ZFS 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 38.

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

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

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

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

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

Le nouveau système de fichiers ZFS, `tank/fs`, peut utiliser autant d'espace disque disponible que nécessaire et est monté automatiquement, à l'adresse suivante : `/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 plus d'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 29.

Création d'un pool de stockage ZFS de base

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

▼ Identification des exigences de stockage du pool de stockage ZFS

1. **Déterminez les périphériques disponibles pour le pool de stockage.**

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

   Pour l'exemple de stockage utilisé dans la section “Création d'un pool de stockage ZFS” à la page 28, partez du principe que les disques entiers `/dev/dsk/c1t0d0` et `/dev/dsk/c2t0d0` sont disponibles.

   Pour plus d'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 33.

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 28 utilise la mise en miroir de base de deux disques disponibles.

Pour plus d'informations sur les fonctions de réplication ZFS, reportez-vous à la section “Fonctions de réplication d'un pool de stockage ZFS” à la page 38.

▼ Création d'un pool de stockage ZFS

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

   Pour plus d'informations sur les profils de droits ZFS, reportez-vous à la section “Profils de droits ZFS” à la page 25.

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

   Le nom sert à identifier le pool de stockage lorsque vous exécutez les commandes `zpool` et `zfs`. Entrez le nom de votre choix, mais celui-ci doit respecter les conventions d'attribution de nom définies dans la section “Exigences d'attribution de noms de composants ZFS” à la page 18.

3. **Créez le pool.**

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

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

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

   Pour plus d'informations sur la création de pools de stockage, reportez-vous à la section “Création de pools de stockage ZFS” à la page 41. Pour plus d'informations sur la détection de l'utilisation de périphériques, reportez-vous à la section “Détection des périphériques utilisés” à la page 50.

4. **Affichez les résultats.**

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

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

   Pour plus d'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 78.

Création d'une hiérarchie de systèmes de fichiers ZFS

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

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

▼ Détermination de la hiérarchie du système de fichiers ZFS

1. **Choisissez la granularité du système de fichiers.**

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

   Deux systèmes de fichiers ZFS, `jeff` et `bill`, sont créés à la section “Création de systèmes de fichiers ZFS” à la page 30.

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

2. **Regroupez les systèmes de fichiers similaires.**

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

   Dans l'exemple de la section “Création de systèmes de fichiers ZFS” à la page 30, 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 30, tous les répertoires personnels sont montés dans `/export/zfs/` *user*. Ils sont partagés à l'aide de NFS et la compression est activée. De plus, un quota de 10 Go est appliqué pour l'utilisateur `jeff`.

Pour plus d'informations sur les propriétés, reportez-vous à la section “Présentation des propriétés ZFS” à la page 139.

▼ Création de systèmes de fichiers ZFS

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

   Pour plus d'informations sur les profils de droits ZFS, reportez-vous à la section “Profils de droits ZFS” à la page 25.

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 share.nfs=on tank/home
   # zfs set compression=on tank/home
   # zfs get compression tank/home
   NAME            PROPERTY       VALUE                    SOURCE
   tank/home       compression    on                       local
   ```

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

   ```
   # zfs create -o mountpoint=/export/zfs -o share.nfs=on -o compression=on tank/home
   ```

   Pour plus d'informations sur les propriétés et l'héritage des propriétés, reportez-vous à “Présentation des propriétés ZFS” à la page 139.

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

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

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

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

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

Pour plus d'informations sur les systèmes de fichiers, reportez-vous à la section “Création d'un système de fichiers ZFS” à la page 136.

Pour plus d'informations sur le montage et le partage de systèmes de fichiers, reportez-vous à la section “Montage de système de fichiers ZFS” à la page 168.

5. Définissez les propriétés spécifiques au système.

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

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

6. Affichez les résultats.

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

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

A noter que l'utilisateur `jeff` a seulement 10 Go de mémoire disponible, tandis que l'utilisateur `bill` peut utiliser le pool entier (67 Go).

Pour plus d'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 160.

Pour plus d'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 20.

♦ ♦ ♦ CHAPITRE 3

Gestion des pools de stockage Oracle Solaris ZFS

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

Ce chapitre contient les sections suivantes :

Composants d'un pool de stockage ZFS

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

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 règle générale, ce périphérique est un disque dur visible pour le système dans le répertoire `/dev/dsk`.

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

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

Part      Tag    Flag     First Sector         Size         Last Sector
0        usr    wm               256       68.36GB          143358320
1 unassigned    wm                 0           0              0
2 unassigned    wm                 0           0              0
3 unassigned    wm                 0           0              0
4 unassigned    wm                 0           0              0
5 unassigned    wm                 0           0              0
6 unassigned    wm                 0           0              0
8   reserved    wm         143358321        8.00MB          143374704
```

Quand Oracle Solaris 11.1 est installé, la plupart du temps une étiquette EFI (GPT) est appliquée aux disques de pool root sur les systèmes x86, similaire à ce qui suit :

```
Current partition table (original):
Total disk sectors available: 27246525 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector        Size        Last Sector
0  BIOS_boot    wm               256     256.00MB         524543
1        usr    wm            524544      12.74GB         27246558
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          27246559       8.00MB         27262942
```

Dans la sortie ci-dessus, la partition 0 (`BIOS boot`) contient des informations d'initialisation GPT requises. Comme la partition 8, elle ne nécessite aucune administration et ne doit pas être modifiée. Le système de fichiers root est contenu dans la partition 1.

Système avec un microprogramme SPARC EFI GPT (prenant en charge les étiquettes) une étiquette de disque EFI est appliquée GPT]. Par exemple :

```
Current partition table (original):
Total disk sectors available: 143358320 + 16384 (reserved sectors)

Part      Tag    Flag     First Sector         Size         Last Sector
0        usr    wm               256       68.36GB          143358320
1 unassigned    wm                 0           0              0
```

```
2 unassigned    wm                0          0          0
3 unassigned    wm                0          0          0
4 unassigned    wm                0          0          0
5 unassigned    wm                0          0          0
6 unassigned    wm                0          0          0
8   reserved    wm        143358321       8.00MB     143374704
```

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

- Avec un disque entier, celui-ci est généralement nommé à l'aide de la convention de nommage `/dev/dsk/cNtNdN`. Certains pilotes tiers suivent une convention de nom différente ou placent les disques à un endroit autre que le répertoire `/dev/dsk`. Pour utiliser ces disques, vous devez les étiqueter manuellement et fournir une tranche à ZFS.
- Sur les systèmes x86, le disque doit avoir une partition `fdisk` Solaris valide. Pour plus d'informations sur la création et la modification d'une partition `fdisk` Solaris, reportez-vous à la section " Configuration de disques pour les systèmes de fichiers ZFS " du manuel " Gestion des périphériques dans Oracle Solaris 11.2 ".
- 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 " Etiquette de disque EFI (GPT) " du manuel " Gestion des périphériques dans Oracle Solaris 11.2 ".
- Le programme d'installation d'Oracle Solaris applique une étiquette EFI (GPT) pour les disques de pool root sur un système SPARC avec un microprogramme compatible GPT et sur un système x86, dans la plupart des cas. Pour plus d'informations, reportez-vous à la section "Configuration requise pour le pool root ZFS" à la page 106.
- Pour des fins de récupération, considérez utiliser la commande `archiveadm` pour créer une archive de pool root. Fractionner la root de pool risque de produire des erreurs car cela exige des étapes manuelles telles que la définition d'un nouveau périphérique d'initialisation, peut-être la mise à jour du fichier `/etc/vfstab` et la réinitialisation d'un fichier de vidage existant.

  Pour plus d'informations sur la création d'une archive de pool root, reportez-vous à " Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ".

Les disques peuvent être spécifiés par le chemin complet comme `/dev/dsk/c1t0d0` ou un nom abrégé composé du nom du périphérique dans le répertoire `/dev/dsk`, par exemple `c1t0d0`. Par exemple, les éléments suivants sont des noms de disques valides :

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

Utilisation de tranches dans un pool de stockage ZFS

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

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

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

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

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

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

Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       1 - 49778       68.36GB    (49778/0/0) 143360640
1 unassigned    wu       0                0         (0/0/0)             0
2     backup    wm       0 - 49778       68.36GB    (49779/0/0) 143363520
3 unassigned    wu       0                0         (0/0/0)             0
4 unassigned    wu       0                0         (0/0/0)             0
5 unassigned    wu       0                0         (0/0/0)             0
```

```
6 unassigned    wu       0               0         (0/0/0)           0
7 unassigned    wu       0               0         (0/0/0)           0
8       boot    wu       0 -     0       1.41MB    (1/0/0)        2880
9 unassigned    wu       0               0         (0/0/0)           0
```

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

Utilisation de fichiers dans un pool de stockage ZFS

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

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

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

Remarques relatives aux pools de stockage ZFS

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

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

fonctionnalités de redondance proposées par la baie. Certaines configurations peuvent fournir une redondance et des performances adéquates, mais d'autres non.

- Vous pouvez construire des périphériques logiques pour ZFS à l'aide des volumes présentés par des gestionnaires de volumes logiciels. Ces configurations sont cependant déconseillées. Même si le système de fichiers ZFS fonctionne correctement sur ces périphériques, il se peut que les performances ne soient pas optimales.

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

- Pour plus d'informations sur les mises en garde concernant les périphériques de pool, reportez-vous à “Modification de périphériques de pool” à la page 307.

Fonctions de réplication d'un pool de stockage ZFS

ZFS assure la redondance des données, ainsi que des propriétés d'auto-rétablissement dans les configurations RAID-Z ou mises en miroir.

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 plus complexe 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 42.

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. RAID-Z à double parité (`raidz2`) est similaire à une RAID-6.

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

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

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

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

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

```
raidz c1t0d0 c2t0d0 c3t0d0
```

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

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

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

Pour obtenir des informations sur les pools de stockage RAID-Z, reportez-vous à la section “Création d'un pool de stockage RAID-Z” à la page 43.

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

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

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

Pool de stockage ZFS hybride

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

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

Données d'autorétablissement dans une configuration redondante

Le système de fichiers ZFS fournit des données d'autorétablissement 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 autre copie redondante et répare les données endommagées en les remplaçant par celles de la copie.

Entrelacement dynamique dans un pool de stockage

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

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

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

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

Création et destruction de pools de stockage ZFS

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

- “Création de pools de stockage ZFS” à la page 41
- “Affichage des informations d'un périphérique virtuel de pool de stockage” à la page 48
- “Gestion d'erreurs de création de pools de stockage ZFS” à la page 50
- “Destruction de pools de stockage ZFS” à la page 53

Création et la destruction de pools sont rapides et faciles. 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. Cette commande simple a des conséquences considérables.

Création de pools de stockage ZFS

Pour créer un pool de stockage, exécutez la commande `zpool create`. Cette commande prend un nom de pool et un nombre illimité de périphériques virtuels en tant qu'arguments. Le nom du pool doit respecter les conventions d'attribution de nom définies à la section “Exigences d'attribution de noms de composants ZFS” à la page 18.

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 deuxième `miroir`indique qu'un nouveau périphérique virtuel de niveau supérieur est spécifié. Les données sont dynamiquement entrelacées sur les deux miroirs, ce qui les rend redondantes sur chaque disque.

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

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

- Ajout d'un autre jeu de disques comme périphérique virtuel (vdev) supplémentaire de niveau supérieur à une configuration en miroir existante. Pour plus d'informations, reportez-vous à la rubrique “Ajout de périphériques à un pool de stockage” à la page 54.
- 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 59.
- 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 67.
- 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 59.

- 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 61.

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

Création d'un pool root ZFS

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

- Oracle Solaris : dans les disques OSOL utilisés pour le pool root sont installés avec une étiquette EFI (GPT) sur un système x86, un système GPT prenant en charge les étiquettes pris en charge du microprogramme SPARC, ou avec une étiquette SMI (VTOC) est appliqué à un système basé sur SPARC GPT prenant en charge du microprogramme sans. Le programme d'installation applique une étiquette EFI (GPT) si possible. Si vous avez besoin de recréer un pool root ZFS après l'installation, vous pouvez utiliser la commande pour appliquer l'étiquette de disque EFI (GPT) et les bonnes informations d'initialisation.

  ```
  # zpool create -B rpool2 c1t0d0
  ```

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

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

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

Pour plus d'informations sur l'installation et l'initialisation d'un système de fichiers root ZFS, reportez-vous au Chapitre 4, Gestion des composants du pool root 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`.

L'exemple suivant montre comment créer 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 parité RAID-Z ou à triple parité configuration d 'à l'aide du mot-clé `raidz2` ou `raidz3` lors de la création du pool. Par exemple :

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

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

errors: No known data errors

# zpool create tank raidz3 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0
c5t0d0 c6t0d0 c7t0d0 c8t0d0
# zpool status -v tank
pool: tank
state: ONLINE
scrub: none requested
config:

NAME        STATE     READ WRITE CKSUM
tank        ONLINE       0     0     0
raidz3-0    ONLINE       0     0     0
c0t0d0      ONLINE       0     0     0
c1t0d0      ONLINE       0     0     0
c2t0d0      ONLINE       0     0     0
c3t0d0      ONLINE       0     0     0
c4t0d0      ONLINE       0     0     0
c5t0d0      ONLINE       0     0     0
c6t0d0      ONLINE       0     0     0
c7t0d0      ONLINE       0     0     0
c8t0d0      ONLINE       0     0     0
errors: No known data errors
```

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

- Ajout d'un autre jeu de disques comme périphérique virtuel (vdev) supplémentaire de niveau supérieur à une configuration RAID-Z existante. Pour plus d'informations, reportez-vous à la rubrique “Ajout de périphériques à un pool de stockage” à la page 54.
- 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 67.

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

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

Pour obtenir des informations supplémentaire, reportez-vous à la section “Configuration de pool de stockage RAID-Z” à la page 39.

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

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

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

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

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

  http://blogs.oracle.com/perrin/entry/slog_blog_or_blogging_on

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

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

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

```
# zpool create datap mirror c0t5000C500335F95E3d0 c0t5000C500335F907Fd0 \
   mirror c0t5000C500335BD117d0 c0t5000C500335DC60Fd0 \
   log mirror c0t5000C500335E106Bd0 c0t5000C500335FC3E7d0

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

NAME                        STATE     READ  WRITE  CKSUM
datap                       ONLINE       0      0      0
   mirror-0                 ONLINE       0      0      0
      c0t5000C500335F95E3d0 ONLINE       0      0      0
      c0t5000C500335F907Fd0 ONLINE       0      0      0
   mirror-1                 ONLINE       0      0      0
      c0t5000C500335BD117d0 ONLINE       0      0      0
      c0t5000C500335DC60Fd0 ONLINE       0      0      0
   logs
   mirror-2                 ONLINE       0      0      0
      c0t5000C500335E106Bd0 ONLINE       0      0      0
      c0t5000C500335FC3E7d0 ONLINE       0      0      0

errors: No known data errors
```

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

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

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

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

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

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

errors: No known data errors
```

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

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

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

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

- L'utilisation de périphériques de cache constitue la meilleure amélioration de performances pour les charges de travail de lecture aléatoire constituées principalement de contenu statique.

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

Précautions pour la création de pools de stockage

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

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

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

Chaque pool de stockage contient un ou plusieurs périphériques virtuels. Un *périphérique virtuel* est une représentation interne du pool de stockage qui décrit la disposition du stockage

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

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

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

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

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

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

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

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

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

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

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

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

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

Gestion d'erreurs de création de pools de stockage ZFS

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

Détection des périphériques utilisés

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

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

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

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

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

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

Elément d'un pool ZFS — Le disque ou fichier fait partie d'un pool de stockage ZFS. Pour corriger cette erreur, utilisez la commande `zpool destroy` afin de détruire l'autre pool s'il est obsolète. Utilisez sinon la commande `zpool detach` pour déconnecter le disque de l'autre pool. Vous pouvez déconnecter un disque que s'il est connecté à un pool de stockage mis en miroir.

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

Contient un système de fichiers	Le disque contient un système de fichiers connu bien qu'il ne soir pas monté et n'apparaisse pas comme étant en cours d'utilisation.
Elément d'un volume	Le disque fait partie d'un volume Solaris Volume Manager.
Elément d'un pool ZFS exporté	Le disque fait partie d'un pool de stockage exporté ou supprimé manuellement d'un système. Dans le deuxième cas, le pool est signalé comme étant `potentiellement actif`, dans la mesure où il peut s'agir d'un disque connecté au réseau en cours d'utilisation par un autre système. Faites attention lorsque vous ignorez un pool potentiellement activé.

L'exemple suivant illustre l'utilisation de l'option `-f` :

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

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

Niveaux de réplication incohérents

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

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

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

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

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

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

tank
  mirror
    c1t0d0
    c1t1d0
```

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

Point de montage par défaut pour les pools de stockage

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

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

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

Pour plus d'informations sur les points de montage, reportez-vous à la section “Gestion des points de montage ZFS” à la page 168.

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 100.

Si vous détruisez un pool à l'aide de la commande `zpool destroy`, le pool reste disponible pour l'importation, comme décrit dans la section “Récupération de pools de stockage ZFS détruits” à la page 100. Cela signifie que des données confidentielles peuvent subsister sur les disques qui faisaient partie du pool. Si vous souhaitez détruire les données placées sur les disques du pool détruit, vous devez utiliser une fonctionnalité telle que l'option `analyze->purge` de l'utilitaire `format` sur tous les disques du pool détruit.

Une autre possibilité pour préserver la confidentialité de données de systèmes de fichiers est de créer des systèmes de fichiers ZFS chiffrés. Lorsqu'un pool contenant un système de fichiers chiffré est détruit, les données ne sont pas accessibles sans les clés de chiffrement, même si le pool détruit est récupéré. Pour plus d'informations, reportez-vous à la section “Chiffrement des systèmes de fichiers ZFS” à la page 191.

Destruction d'un pool avec des périphériques disponibles

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

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

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

```
# zpool destroy -f tank
```

Pour plus d'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 86.

Pour plus d'informations sur l'importation de pools, reportez-vous à la section “Importation de pools de stockage ZFS” à la page 95.

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 33. 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

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

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

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

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

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

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

Pour plus d'informations sur la validation des périphériques virtuels, reportez-vous à la section “Détection des périphériques utilisés” à la page 50.

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

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

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

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

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

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

```
      c0t3d0   ONLINE       0      0      0
      c1t3d0   ONLINE       0      0      0

errors: No known data errors
```

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

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

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

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

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

NAME           STATE     READ  WRITE  CKSUM
rzpool         ONLINE       0      0      0
   raidz1-0    ONLINE       0      0      0
      c1t2d0   ONLINE       0      0      0
      c1t3d0   ONLINE       0      0      0
      c1t4d0   ONLINE       0      0      0
   raidz1-1    ONLINE       0      0      0
      c2t2d0   ONLINE       0      0      0
      c2t3d0   ONLINE       0      0      0
      c2t4d0   ONLINE       0      0      0

errors: No known data errors
```

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

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

```
# 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
```

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

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

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

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

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

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

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

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

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

EXEMPLE 3-4 Ajout et suppression des périphériques de cache

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

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

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

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

errors: No known data errors
```

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

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

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

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

errors: No known data errors
```

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

Pour plus d'informations sur l'utilisation des périphériques de cache dans un pool de stockage ZFS, reportez-vous à la section “Création d'un pool de stockage ZFS avec des périphériques de cache” à la page 47.

Connexion et séparation de périphériques dans un pool de stockage

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

Si vous connectez un disque pour créer un pool root mis en miroir, reportez-vous à la section “Configuration d'un pool root mis en miroir (SPARC ou x86/VTOC)” à la page 111.

Si vous remplacez un disque dans le pool root ZFS, reportez-vous à la section “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/VTOC)” à la page 115.

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

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

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

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

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

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

errors: No known data errors
```

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

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

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

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

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

```
pool: tank
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Fri Jan  8 14:28:23 2010
config:

NAME           STATE     READ  WRITE  CKSUM
tank           ONLINE       0      0      0
   mirror-0    ONLINE       0      0      0
      c0t1d0   ONLINE       0      0      0
      c1t1d0   ONLINE       0      0      0  73.5K resilvered

errors: No known data errors
```

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

```
# zpool detach zeepool c2t1d0
```

Toutefois, cette opération échouera en l'absence de répliques de données valides. Par exemple :

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

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

Un pool de stockage ZFS mis en miroir peut être rapidement cloné en tant que pool de sauvegarde à l'aide de la commande `zpool split`.

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

Remarque - Pour consulter d'autres procédures ZFS et ainsi que des exemples illustrant le fractionnement d'un pool moyennant la commande`zpool split`, connectez-vous à votre compte My Oracle Support (https://support.oracle.com) et à la section *Utilisation de 'zpools split' pour fractionner un rpool (Doc ID 1637715.1)*.

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

```
# zpool status tank
pool: tank
```

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

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

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

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

errors: No known data errors

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

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

errors: No known data errors
```

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

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

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

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

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

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

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

- Cette fonction n'est pas disponible dans une configuration RAID-Z ou un pool non redondant composé de plusieurs disques.
- Avant de tenter une opération `zpool split`, les opérations des données et des applications doivent être suspendues.
- Vous ne pouvez pas scinder un pool si une réargenture est en cours.
- Lorsqu'un pool mis en miroir est composé de deux à trois disques et que le dernier disque du pool d'origine est utilisé pour le nouveau pool créé, la meilleure solution consiste à scinder le pool mis en miroir. Vous pouvez ensuite utiliser la commande `zpool attach` pour recréer votre pool de stockage d'origine mis en miroir ou convertir votre nouveau pool dans un pool de stockage mis en miroir. Aucune méthode ne permet de créer un *nouveau* pool mis en miroir à partir d'un pool mis en miroir *existant* en une opération `zpool split`, car le nouveau pool (séparé) n'est pas redondant.
- Si le pool existant est un miroir tridirectionnel, le nouveau pool contiendra un disque après l'opération de scission. Si le pool existant est un miroir bidirectionnel composé de deux disques, cela donne deux pools non redondants composés de deux disques. Vous devez connecter deux disques supplémentaires pour convertir les pools non redondants en pools mis en miroir.
- Pour conserver vos données redondantes lors d'une scission, scindez un pool de stockage mis en miroir composé de trois disques pour que le pool d'origine soit composé de deux disques mis en miroir après la scission.
- Confirmez que votre matériel est configuré correctement avant de séparer un pool mis en miroir. Pour plus d'informations sur la confirmation des paramètres de purge du cache de votre matériel, reportez-vous à la section “Pratiques recommandées générales” à la page 321.

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

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

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

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

```
scan: none requested
config:

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

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

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

errors: No known data errors

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

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

errors: No known data errors
```

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

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

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

Mise hors ligne d'un périphérique

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

```
# zpool offline tank c0t5000C500335F95E3d0
```

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

- Vous ne pouvez pas mettre un périphérique hors ligne au point où il devient `UNAVAIL`. Par exemple, vous ne pouvez pas mettre hors ligne deux périphériques dans une configuration `raidz1`, ni mettre hors ligne un périphérique virtuel de niveau supérieur.

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

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

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

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

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

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

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

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

  Une autre mode d'utilisation d'un périphérique provenant d'un autre pool de stockage si vous souhaitez conserver le pool de stockage d'origine consiste à remplacer le périphérique existant dans le pool de stockage d'origine par un autre périphérique similaire. Pour obtenir des informations sur le remplacement de périphériques, reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 67.

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 78.

Pour plus d'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 86.

Mise en ligne d'un périphérique

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

```
# zpool online tank c0t5000C500335F95E3d0
```

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

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

```
# zpool online tank c0t5000C500335DC60Fd0
warning: device 'c0t5000C500335DC60Fd0' onlined, but remains in faulted state
use 'zpool clear' to restore a faulted device
```

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

```
SUNW-MSG-ID: ZFS-8000-LR, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Wed Jun 20 11:35:26 MDT 2012
PLATFORM: ORCL,SPARC-T3-4, CSN: 1120BDRCCD, HOSTNAME: tardis
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: fb6699c8-6bfb-eefa-88bb-81479182e3b7
DESC: ZFS device 'id1,sd@n5000c500335dc60f/a' in pool 'pond' failed to open.
AUTO-RESPONSE: An attempt will be made to activate a hot spare if available.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this event.
Run 'zpool status -lx' for more information. Please refer to the associated
reference document at http://support.oracle.com/msg/ZFS-8000-LR for the latest
service procedures and policies regarding this diagnosis.
```

Pour obtenir des informations sur le remplacement d'un périphérique défaillant, reportez-vous à la section “Résolution d'un périphérique manquant ou supprimé” à la page 292.

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 `autoexpand` du pool ne soit activée. Vous pouvez étendre automatiquement le LUN à l'aide de la commande `zpool online-e`, même si le LUN est déjà en ligne ou s'il est actuellement hors ligne. Par exemple :

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

Effacement des erreurs de périphérique de pool de stockage

Si un périphérique est mis hors ligne en raison d'une défaillance qui entraîne l'affichage d'erreurs dans la sortie `zpool status`, vous pouvez effacer les décomptes d'erreurs à l'aide de la commande `zpool clear`. Si un périphérique perd sa connexion au sein d'un pool, puis si la connectivité est restaurée, vous devrez également supprimer ces erreurs.

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

```
# zpool clear tank
```

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

```
# zpool clear tank c0t5000C500335F95E3d0
```

Pour de plus amples informations sur l'effacement d'erreurs de `zpool`, reportez-vous à la section “Erreurs de périphérique Non persistant ou Persistant de compensation” à la page 298.

Remplacement de périphériques dans un pool de stockage

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

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

```
# zpool replace tank c1t1d0
```

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

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

Si vous remplacez un disque dans le pool root ZFS, reportez-vous à la section “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/VTOC)” à la page 115.

Voici les étapes de base pour remplacer un disque :

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

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

   ```
   # zpool replace tank c1t1d0
   ```

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

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

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

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

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

Pour consulter un exemple de remplacement d'un disque SATA, reportez-vous à l'Exemple 10-1, “Remplacement d'un disque SATA dans un pool de stockage ZFS”.

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

- Si vous définissez la propriété de pool `autoreplace` sur `on`, tout nouveau périphérique détecté au même emplacement physique qu'un périphérique appartenant précédemment au pool est automatiquement formaté et remplacé. Lorsque cette propriété est activée, vous n'êtes pas obligé d'utiliser la commande `zpool replace`. Cette fonction n'est pas disponible sur tous les types de matériel.
- L'état de pool de stockage `REMOVED` est fourni en cas de retrait physique du périphérique ou d'un disque hot spare alors que le système est en cours d'exécution. Si un disque hot spare est disponible, il remplace le périphérique retiré.
- Si un périphérique est retiré, puis réinséré, il est mis en ligne. Si un disque hot spare est activé lors de la réinsertion du périphérique, le disque hot spare est retiré une fois l'opération en ligne terminée.
- La détection automatique du retrait ou de l'insertion de périphériques dépend du matériel utilisé. Il est possible qu'elle ne soit pas prise en charge sur certaines plates-formes. Par exemple, les périphériques USB sont configurés automatiquement après insertion. Il peut

être toutefois nécessaire d'utiliser la commande `cfgadm -c configure` pour configurer un lecteur SATA.

- Les disques hot spare sont consultés régulièrement afin de vérifier qu'ils sont en ligne et disponibles.
- La taille du périphérique de remplacement doit être égale ou supérieure au disque le plus petit d'une configuration RAID-Z ou mise en miroir.
- Lorsqu'un périphérique de remplacement dont la taille est supérieure à la taille du périphérique qu'il remplace est ajouté à un pool, ce dernier n'est pas automatiquement étendu à sa taille maximale. La valeur de 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. Assurez-vous qu'une réargenture complète a été effectuée sur le premier périphérique avant de tenter le remplacement du deuxième périphérique. La propriété `autoexpand` est activée après les remplacements de disque pour étendre le disque à sa taille maximale.

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

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

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

- Le remplacement d'un grand nombre de disques dans un pool volumineux prend du temps, en raison de la réargenture des données sur les nouveaux disques. En outre, il peut s'avérer

utile d'exécuter la commande `zpool scrub` entre chaque remplacement de disque afin de garantir le fonctionnement des périphériques de remplacement et l'exactitude des données écrites.

- Si un disque défectueux a été remplacé automatiquement par un disque hot spare, il se peut que vous deviez déconnecter le disque hot spare une fois le disque défectueux remplacé. Vous pouvez utiliser la commande `zpool detach` pour déconnecter le disque hot spare d'un pool RAID-Z ou mis en miroir. Pour plus d'informations sur la déconnexion d'un disque hot spare, reportez-vous à la section “Activation et désactivation de disques hot spare dans le pool de stockage” à la page 72.

Pour plus d'informations sur le remplacement de périphériques, reportez-vous aux sections “Résolution d'un périphérique manquant ou supprimé” à la page 292 et “Remplacement ou réparation d'un périphérique endommagé ” à la page 296.

Désignation des disques hot spare dans le pool de stockage

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

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

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

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

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

NAME                        STATE     READ  WRITE  CKSUM
zeepool                     ONLINE       0      0      0
   mirror-0                 ONLINE       0      0      0
      c0t5000C500335F95E3d0 ONLINE       0      0      0
      c0t5000C500335F907Fd0 ONLINE       0      0      0
```

```
   mirror-1                  ONLINE       0      0      0
      c0t5000C500335BD117d0  ONLINE       0      0      0
      c0t5000C500335DC60Fd0  ONLINE       0      0      0
   spares
      c0t5000C500335E106Bd0    AVAIL
      c0t5000C500335FC3E7d0    AVAIL

errors: No known data errors
```

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

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

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

errors: No known data errors
```

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

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

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

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

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

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

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

  `cannot replace` *disk3* `with` *disk4*`: device is too small`
- Vous ne pouvez pas partager un disque de rechange entre plusieurs systèmes.

  Vous ne pouvez pas configurer plusieurs systèmes de manière à ce qu'ils partagent un disque de rechange, même si ce disque apparaît comme accessible aux systèmes concernés. Si un disque est configuré pour être partagé par plusieurs pools, il faut que ces pools soient contrôlés par un système unique.
- Tenez compte du fait que si vous partagez un disque de rechange entre deux pools de données sur le même système, vous devez coordonner l'utilisation du disque de rechange entre les deux pools. Par exemple, le pool A utilise le disque de rechange et est exporté. Sans le savoir, le pool B peut utiliser le disque de rechange lors de l'exportation du pool A. Lorsque le pool A est importé, une altération des données peut se produire car les deux pools utilisent le même disque. Par conséquent, soyez attentif à ces situations marginales où des problèmes peuvent se poser pour les pools dans certaines circonstances et ce, même si un disque de rechange est partagé par plusieurs pools.
- Ne partagez pas un disque hot spare entre un pool root et un pool de données.

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

Les disques hot spare s'activent des façons suivantes :

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

  En cas de défaillance d'un disque hot spare en cours d'utilisation, l'agent FMA sépare le disque hot spare et annule ainsi le remplacement. L'agent tente ensuite de remplacer le

périphérique par un autre disque hot spare s'il y en a un de disponible. Cette fonction est actuellement limitée par le fait que le moteur de diagnostics ZFS ne génère des défaillances qu'en cas de disparition d'un périphérique du système.

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

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

```
# zpool status -x
pool: zeepool
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
Run 'zpool status -v' to see device specific details.
scan: resilvered 3.15G in 0h0m with 0 errors on Thu Jun 21 16:46:19 2012
config:


NAME                        STATE     READ  WRITE  CKSUM
zeepool                     ONLINE       0      0      0
   mirror-0                 ONLINE       0      0      0
      c0t5000C500335F95E3d0 ONLINE       0      0      0
      c0t5000C500335F907Fd0 ONLINE       0      0      0
   mirror-1                 DEGRADED     0      0      0
      c0t5000C500335BD117d0 ONLINE       0      0      0
   spare-1                  DEGRADED   449      0      0
      c0t5000C500335DC60Fd0 UNAVAIL      0      0      0
      c0t5000C500335E106Bd0 ONLINE       0      0      0
   spares
      c0t5000C500335E106Bd0      INUSE

errors: No known data errors
```

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

- Suppression du disque hot spare du pool de stockage.
- Déconnexion du disque hot spare après avoir remplacé physiquement un disque défectueux. Reportez-vous à l'Exemple 3-8, “Déconnexion d'un disque hot spare après le remplacement du disque défectueux”.
- Remplacement temporaire ou permanent par un autre disque hot spare. Reportez-vous à la section Exemple 3-9, “Déconnexion d'un disque défectueux et utilisation d'un disque hot spare”.

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

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

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

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

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

```
# zpool detach zeepool c0t5000C500335E106Bd0
```

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

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

```
# zpool status zeepool
pool: zeepool
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
Run 'zpool status -v' to see device specific details.
scan: scrub in progress since Thu Jun 21 17:01:49 2012
1.07G scanned out of 6.29G at 220M/s, 0h0m to go
0 repaired, 17.05% done
config:
NAME                        STATE     READ  WRITE  CKSUM
zeepool                     ONLINE       0      0      0
   mirror-0                 ONLINE       0      0      0
```

```
      c0t5000C500335F95E3d0   ONLINE       0      0      0
      c0t5000C500335F907Fd0   ONLINE       0      0      0
   mirror-1                   DEGRADED     0      0      0
      c0t5000C500335BD117d0   ONLINE       0      0      0
      c0t5000C500335DC60Fd0   UNAVAIL      0      0      0
   spares
      c0t5000C500335E106Bd0    AVAIL

errors: No known data errors
# zpool detach zeepool c0t5000C500335DC60Fd0
# zpool status zeepool
pool: zeepool
state: ONLINE
scan: resilvered 3.15G in 0h0m with 0 errors on Thu Jun 21 17:02:35 2012
config:

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

errors: No known data errors
(Original failed disk c0t5000C500335DC60Fd0 is physically replaced)
# zpool add zeepool spare c0t5000C500335DC60Fd0
# zpool status zeepool
pool: zeepool
state: ONLINE
scan: resilvered 3.15G in 0h0m with 0 errors on Thu Jun 21 17:02:35 2012
config:

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

errors: No known data errors
```

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

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

Gestion des propriétés de pool de stockage ZFS

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

```
# zpool get all zeepool
NAME     PROPERTY       VALUE                SOURCE
zeepool  allocated      6.29G                -
zeepool  altroot        -                    default
zeepool  autoexpand     off                  default
zeepool  autoreplace    off                  default
zeepool  bootfs         -                    default
zeepool  cachefile      -                    default
zeepool  capacity       1%                   -
zeepool  dedupditto     0                    default
zeepool  dedupratio     1.00x                -
zeepool  delegation     on                   default
zeepool  failmode       wait                 default
zeepool  free           550G                 -
zeepool  guid           7543986419840620672  -
zeepool  health         ONLINE               -
zeepool  listshares     off                  default
zeepool  listsnapshots  off                  default
zeepool  readonly       off                  -
zeepool  size           556G                 -
zeepool  version        34                   default
```

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

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

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

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

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

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

Nom de la propriété	Type	Valeur par défaut	Description
`alloué`	String	S/O	Valeur en lecture seule indiquant la quantité d'espace de stockage contenu dans le pool qui a été physiquement alloué.
`altroot`	String	`off`	Identifie un répertoire root alternatif. S'il est défini, ce répertoire est ajouté au début de tout point de montage figurant dans le pool. Cette propriété peut être utilisée lors de l'examen d'un pool

Nom de la propriété	Type	Valeur par défaut	Description
			inconnu si les points de montage ne sont pas fiables ou dans un environnement d'initialisation alternatif dans lequel les chemins types sont incorrects.
`autoreplace`	Valeur booléenne	`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 `zpoolreplace`. Si la valeur est définie sur `on`, tout nouveau périphérique se trouvant au même emplacement physique qu'un périphérique qui appartenait au pool est automatiquement formaté et remplacé. L'abréviation de la propriété est la suivante : `replace`.
`bootfs`	Valeur booléenne	S/O	Identifie le système de fichiers amorçable par défaut pour le pool root. Cette propriété est généralement définie par les programmes d'installation.
`cachefile`	String	S/O	Contrôle où la configuration du pool est mise en cache. Tous les pools du cache sont importés automatiquement à l'initialisation du système. Toutefois, dans les environnements d'installation et de clustering, il peut s'avérer nécessaire de placer ces informations en cache à un autre endroit afin d'éviter l'importation automatique des pools. Vous pouvez définir cette propriété pour mettre en cache les informations de configuration du pool dans un autre emplacement. Ces informations peuvent être importées ultérieurement à l'aide de la commande `zpool import-c`. Pour la plupart des configurations ZFS, cette propriété n'est pas utilisée.
`Capacité`	Nombre	S/O	Valeur de capacité en lecture seule identifiant le pourcentage d'espace utilisé par le pool. L'abréviation de la propriété est `cap`.
`dedupditto`	String	S/O	Définit un seuil; si le nombre de références pour un bloc dédupliqué dépasse ce seuil, une autre copie ditto du bloc est automatiquement stockée.
`dedupratio`	String	S/O	Dedupratio ratio en lecture seule obtenu pour un pool, exprimé sous la forme d'un multiple.
`delegation`	Valeur booléenne	`on`	Contrôle si un utilisateur non privilégié peut bénéficier des autorisations d'accès définies pour un système de fichiers. Pour plus d'informations, reportez-vous au Chapitre 8, Administration déléguée de ZFS dans Oracle Solaris.
`failmode`	String	`wait`	Contrôle le comportement du système en cas de panne grave d'un pool. Cette condition résulte habituellement d'une perte de connectivité aux périphériques de stockage sous-jacents ou d'une panne de tous les périphériques au sein du pool. Le comportement d'un événement de ce type est déterminé par l'une des valeurs suivantes : ■ `wait` : bloque toutes les demandes d'E/S vers le pool jusqu'au rétablissement de la connectivité et jusqu'à l'effacement des erreurs à l'aide de la commande `zpool clear`. Dans cet état, les opérations d'E/S du pool sont bloquées mais les opérations de lecture peuvent aboutir. Un pool renvoie l'état `wait` jusqu'à ce que le problème du périphérique soit résolu. ■ `continue` : renvoie une erreur EIO à toute nouvelle demande d'E/S d'écriture, mais autorise les lectures de tout autre

Nom de la propriété	Type	Valeur par défaut	Description
			périphérique fonctionnel. Toute demande d'écriture devant encore être validée sur disque est bloquée. Une fois le périphérique reconnecté ou remplacé, les erreurs doivent être effacées à l'aide de la commande `zpool clear`. ■ `panic` : affiche un message sur la console et génère un vidage sur incident du système.
`free`	String	S/O	Valeur en lecture seule identifiant le nombre de blocs non alloués au sein du pool.
`guid`	String	S/O	Propriété en lecture seule identifiant l'identificateur unique pour le pool.
`Intégrité`	String	S/O	Propriété en lecture seule indiquant l'état actuel du pool comme ONLINE, DEGRADED, SUSPENDED, REMOVED ou UNAVAIL.
`listshares`	String	arrêté	Contrôle si les informations partagées dans ce pool sont affichées à l'aide de la commande `zfs list`. La valeur par défaut est `off`.
`listsnapshots`	String	arrêté	Détermine si les informations qui lui sont associées instantané ce pool sont affichées à l'aide de la commande `zfs list`. Si cette propriété est désactivée, vous pouvez afficher les informations relatives à instantané et la commande`zfs list -t snapshot`.
`readonly`	Valeur booléenne	arrêté	Indique si un pool peut être modifié. Cette propriété est uniquement activée lorsqu'un pool a été importé en mode lecture seule. Lorsqu'elle est activée, les données synchrones éventuellement présentes dans le journal d'intention ne sont pas accessibles tant que le pool n'a pas réimporté en mode lecture-écriture.
`Taille`	Nombre	S/O	Propriété en lecture seule identifiant la taille totale du pool de stockage.
`version`	Nombre	S/O	Identifie la version actuelle sur disque du pool. La méthode recommandée de mise à jour des pools consiste à utiliser la commande `zpool upgrade`, bien que cette propriété puisse être utilisée lorsqu'une version spécifique est requise pour des raisons de compatibilité ascendante. Cette propriété peut être définie sur n'importe quel nombre compris entre 1 et la version actuelle signalée par l'option v commande. `zpool upgrade`-

Requête d'état de pool de stockage ZFS

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

- “Affichage des informations des pools de stockage ZFS” à la page 79
- “Visualisation des statistiques d'E/S des pools de stockage ZFS” à la page 83
- “Détermination de l'état de maintenance des pools de stockage ZFS” à la page 86

Affichage des informations des pools de stockage ZFS

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

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

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

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

La sortie de cette commande affiche les informations suivantes :

`NAME` Nom du pool.

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

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

Pour plus d'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 20.

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

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

`HEALTH` Etat de maintenance actuel du pool.

Pour plus d'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 86.

`ALTROOT` Root de remplacement, le cas échéant.

Pour plus d'informations sur les pools root de remplacement, reportez-vous à “Utilisation d'un pool ZFS avec un emplacement de root de remplacement” à la page 279.

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

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

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

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

Affichage des périphériques de pool par emplacement physique

Vous pouvez utiliser l'option `zpool status -l` pour afficher des informations sur l'emplacement physique des périphériques de pool. Les informations sur l'emplacement physique sont utiles si vous devez supprimer ou remplacer un disque physiquement.

En outre, vous pouvez utiliser la commande `fmadm add-alias` pour inclure un nom d'alias de disque qui facilite l'identification de l'emplacement physique des disques dans votre environnement. Par exemple :

```
# fmadm add-alias SUN-Storage-J4400.1002QCQ015 Lab10Rack5...
```

```
# zpool status -l tank
pool: tank
state: ONLINE
scan: scrub repaired 0 in 0h0m with 0 errors on Fri Aug  3 16:00:35 2012
config:

NAME                                          STATE     READ  WRITE  CKSUM
tank                                          ONLINE       0      0      0
   mirror-0                                   ONLINE       0      0      0
      /dev/chassis/Lab10Rack5.../DISK_02/disk   ONLINE       0      0      0
      /dev/chassis/Lab10Rack5.../DISK_20/disk   ONLINE       0      0      0
   mirror-1                                   ONLINE       0      0      0
      /dev/chassis/Lab10Rack5.../DISK_22/disk   ONLINE       0      0      0
      /dev/chassis/Lab10Rack5.../DISK_14/disk   ONLINE       0      0      0
   mirror-2                                   ONLINE       0      0      0
      /dev/chassis/Lab10Rack5.../DISK_10/disk   ONLINE       0      0      0
      /dev/chassis/Lab10Rack5.../DISK_16/disk   ONLINE       0      0      0
   mirror-3                                   ONLINE       0      0      0
```

```
      /dev/chassis/Lab10Rack5.../DISK_01/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_21/disk   ONLINE       0     0     0
   mirror-4                                     ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_23/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_15/disk   ONLINE       0     0     0
   mirror-5                                     ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_09/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_04/disk   ONLINE       0     0     0
   mirror-6                                     ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_08/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_05/disk   ONLINE       0     0     0
   mirror-7                                     ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_07/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_11/disk   ONLINE       0     0     0
   mirror-8                                     ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_06/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_19/disk   ONLINE       0     0     0
   mirror-9                                     ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_00/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_13/disk   ONLINE       0     0     0
   mirror-10                                    ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_03/disk   ONLINE       0     0     0
      /dev/chassis/Lab10Rack5.../DISK_18/disk   ONLINE       0     0     0
   spares
      /dev/chassis/Lab10Rack5.../DISK_17/disk   AVAIL
      /dev/chassis/Lab10Rack5.../DISK_12/disk   AVAIL

errors: No known data errors
```

Affichage de statistiques spécifiques à un pool de stockage

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

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

Les noms de colonne correspondent aux propriétés répertoriées à la section “Affichage des informations concernant tous les pools de stockage ou un pool spécifique” à la page 79.

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. Permet d'obtenir la liste des noms de pool du système, vous devez utiliser la syntaxe suivante :

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

Voici un autre exemple :

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

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

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

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

```
# zpool history
History for 'rpool':
2012-04-06.14:02:55 zpool create -f rpool c3t0d0s0
2012-04-06.14:02:56 zfs create -p -o mountpoint=/export rpool/export
2012-04-06.14:02:58 zfs set mountpoint=/export rpool/export
2012-04-06.14:02:58 zfs create -p rpool/export/home
2012-04-06.14:03:03 zfs create -p -V 2048m rpool/swap
2012-04-06.14:03:08 zfs set primarycache=metadata rpool/swap
2012-04-06.14:03:09 zfs create -p -V 4094m rpool/dump
2012-04-06.14:26:47 zpool set bootfs=rpool/ROOT/s11u1 rpool
2012-04-06.14:31:15 zfs set primarycache=metadata rpool/swap
2012-04-06.14:31:46 zfs create -o canmount=noauto -o mountpoint=/var/share rpool/VARSHARE
2012-04-06.15:22:33 zfs set primarycache=metadata rpool/swap
2012-04-06.16:42:48 zfs set primarycache=metadata rpool/swap
2012-04-09.16:17:24 zfs snapshot -r rpool/ROOT@yesterday
2012-04-09.16:17:54 zfs snapshot -r rpool/ROOT@now
```

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

Les caractéristiques de l'historique sont les suivantes :

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

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

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

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

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

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

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

```
# zpool history -i tank
History for 'tank':
2012-01-25.16:35:32 zpool create -f tank mirror c3t1d0 c3t2d0 spare c3t3d0
2012-01-25.16:35:32 [internal pool create txg:5] pool spa 33; zfs spa 33; zpl 5;
uts tardis 5.11 11.1 sun4v
2012-02-17.13:04:10 zfs create tank/test
2012-02-17.13:04:10 [internal property set txg:66094] $share2=2 dataset = 34
2012-02-17.13:04:31 [internal snapshot txg:66095] dataset = 56
2012-02-17.13:05:01 zfs snapshot -r tank/test@snap1
2012-02-17.13:08:00 [internal user hold txg:66102] <.send-4736-1> temp = 1 ...
```

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

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

`alloc capacity` Capacité utilisée, c'est-à-dire quantité de données actuellement stockées dans le pool ou le périphérique. Cette quantité diffère quelque peu de la

quantité d'espace disque disponible pour les systèmes de fichiers effectifs en raison de détails d'implémentation interne.

Pour plus d'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 20.

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

`read operations` — Nombre d'opérations de lecture d'E/S envoyées au pool ou au périphérique, y compris les demandes de métadonnées.

`write operations` — Nombre d'opérations d'écriture d'E/S envoyées au pool ou au périphérique.

`read bandwidth` — Bande passante de toutes les opérations de lecture (métadonnées incluses), exprimée en unités par seconde.

`write bandwidth` — Bande passante de toutes les opérations d'écriture, exprimée en unités par seconde.

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

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

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

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

```
# zpool iostat tank 2
capacity     operations    bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
tank        18.5G  49.5G      0    187      0  23.3M
tank        18.5G  49.5G      0    464      0  57.7M
```

```
tank        18.5G  49.5G      0    457      0  56.6M
tank        18.8G  49.2G      0    435      0  51.3M
```

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

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

Liste des statistiques d'E/S des périphériques virtuels

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

```
# zpool iostat -v
capacity     operations    bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
rpool       6.05G  61.9G      0      0    785    107
mirror      6.05G  61.9G      0      0    785    107
c1t0d0s0        -      -      0      0    578    109
c1t1d0s0        -      -      0      0    595    109
----------  -----  -----  -----  -----  -----  -----
tank        36.5G  31.5G      4      1   295K   146K
mirror      36.5G  31.5G    126     45  8.13M  4.01M
c1t2d0          -      -      0      3   100K   386K
c1t3d0          -      -      0      3   104K   386K
----------  -----  -----  -----  -----  -----  -----
```

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

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

création du pool, qui n'est pas comptabilisée au niveau du miroir. Ces chiffres s'égalisent graduellement dans le temps. Cependant, les périphériques hors ligne, ne répondant pas, ou en panne peuvent également affecter cette symétrie.

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

En outre, vous pouvez afficher des informations sur l'emplacement physique des périphériques virtuels du pool. Par exemple :

```
# zpool iostat -lv
capacity     operations    bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
export      2.39T  2.14T     13     27  42.7K   300K
mirror     490G   438G      2      5  8.53K  60.3K
/dev/chassis/lab10rack15/SCSI_Device__2/disk      -      -      1      0  4.47K  60.3K
/dev/chassis/lab10rack15/SCSI_Device__3/disk      -      -      1      0  4.45K  60.3K
mirror     490G   438G      2      5  8.62K  59.9K
/dev/chassis/lab10rack15/SCSI_Device__4/disk      -      -      1      0  4.52K  59.9K
/dev/chassis/lab10rack15/SCSI_Device__5/disk      -      -      1      0  4.48K  59.9K
mirror     490G   438G      2      5  8.60K  60.2K
/dev/chassis/lab10rack15/SCSI_Device__6/disk      -      -      1      0  4.50K  60.2K
/dev/chassis/lab10rack15/SCSI_Device__7/disk      -      -      1      0  4.49K  60.2K
mirror     490G   438G      2      5  8.47K  60.1K
/dev/chassis/lab10rack15/SCSI_Device__8/disk      -      -      1      0  4.42K  60.1K
/dev/chassis/lab10rack15/SCSI_Device__9/disk      -      -      1      0  4.43K  60.1K
.
.
.
```

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

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

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

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

DEGRADED

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

ONLINE

Pool dont tous les périphériques fonctionnent normalement.

SUSPENDED

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

UNAVAIL

Pool avec des métadonnées endommagées, ou des périphériques non disponibles, et pas assez de répliques pour continuer de fonctionner.

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

DEGRADED	Le périphérique virtuel a connu un panne. Toutefois, il continue de fonctionner. Cet état est le plus commun lorsqu'un miroir ou un périphérique RAID-Z a perdu un ou plusieurs périphériques le constituant. La tolérance de pannes du pool peut être compromise dans la mesure où une défaillance ultérieure d'un autre périphérique peut être impossible à résoudre.
OFFLINE	Le périphérique a été mis hors ligne explicitement par l'administrateur.
ONLINE	Le périphérique ou le périphérique virtuel fonctionne normalement. Même si certaines erreurs transitoires peuvent encore survenir, le périphérique fonctionne correctement.
REMOVED	Le périphérique a été retiré alors que le système était en cours d'exécution. La détection du retrait d'un périphérique dépend du matériel et n'est pas pris en charge sur toutes les plates-formes.
UNAVAIL	L'ouverture du périphérique ou du périphérique virtuel est impossible. Dans certains cas, les pools avec des périphériques en état UNAVAIL s'affichent en mode DEGRADED. Si un périphérique de niveau supérieur est en état UNAVAIL, aucun élément du pool n'est accessible.

La maintenance d'un pool est déterminée à partir de celle de l'ensemble de ses périphériques virtuels. Si l'état de tous les périphériques virtuels est ONLINE, l'état du pool est également ONLINE. Si l'état d'un des périphériques virtuels est DEGRADED ou UNAVAIL, l'état du pool est également DEGRADED. Si l'état d'un des périphériques virtuels est UNAVAIL ou OFFLINE, l'état du pool est également UNAVAIL ou SUSPENDED. Un pool en état UNAVAIL ou SUSPENDED est complètement inaccessible. Aucune donnée ne peut être récupérée tant que les périphériques nécessaires n'ont pas été connectés ou réparés. Un pool renvoyant l'état DEGRADED continue

à être exécuté. Cependant, il se peut que vous ne puissiez pas atteindre le même niveau de redondance ou de capacité de données que s'il se trouvait en ligne.

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

- Rapport de progression de la réargenture. Par exemple :

```
scan: resilver in progress since Wed Jun 20 14:19:38 2012
7.43G scanned
7.43G resilvered at 26.8M/s, 10.35% done, 0h30m to go
```

- Rapport de progression du nettoyage. Par exemple :

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

- Message de fin de la réargenture. Par exemple :

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

- Message de fin du nettoyage. Par exemple :

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

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

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

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

Etat de maintenance de base de pool de stockage

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

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

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

Etat de maintenance détaillé

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

```
# zpool status -v pond
```

```
pool: pond
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
scan: scrub repaired 0 in 0h0m with 0 errors on Wed Jun 20 15:38:08 2012
config:

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

device details:

c0t5000C500335F907Fd0    UNAVAIL          cannot open
status: ZFS detected errors on this device.
The device was missing.
see: http://support.oracle.com/msg/ZFS-8000-LR for recovery


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, y compris une description lisible du problème et un lien vers un article de la base de connaissances pour plus d'informations. Les articles de connaissances donnent les informations les plus récentes vous permettant de résoudre le problème. Les informations détaillées de configuration doivent vous permettre de déterminer les périphériques endommagés et la manière de réparer le pool.

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

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

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

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

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

```
# zpool status -x
pool: pond
state: DEGRADED
status: One or more devices has been taken offline by the administrator.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Online the device using 'zpool online' or replace the device with
'zpool replace'.
config:

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

errors: No known data errors
```

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

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

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

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

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

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

```
# zpool status -T d 3 2
Wed Jun 20 16:10:09 MDT 2012
pool: pond
state: ONLINE
scan: resilvered 9.50K in 0h0m with 0 errors on Wed Jun 20 16:07:34 2012
config:
```

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

errors: No known data errors

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

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

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

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

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

errors: No known data errors

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

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

errors: No known data errors
```

Migration de pools de stockage ZFS

Parfois, il est possible que vous deviez déplacer un pool de stockage d'un système à l'autre. Pour ce faire, les périphériques de stockage doivent être déconnectés du système d'origine et reconnectés au système de destination. Pour accomplir cette tâche, vous pouvez raccorder physiquement les périphériques ou utiliser des périphériques multiport, par exemple les périphériques d'un SAN. Le système de fichiers ZFS permet d'exporter le pool à partir d'un système et de l'importer sur le système de destination, même si l'endianisme de l'architecture des systèmes est différente. Pour plus d'informations sur la réplication ou la migration de systèmes de fichiers entre un pool de stockage à un autre peuvent se trouver sur des systèmes différents, reportez-vous à la section “Envoi et réception de données ZFS” à la page 213.

Préparatifs de migration de pool de stockage ZFS

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

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

Exportation d'un pool de stockage ZFS

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

```
# zpool export tank
```

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

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

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

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

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

Pour plus d'informations sur les volumes ZFS, reportez-vous à la section “Volumes ZFS” à la page 269.

Définition des pools de stockage disponibles pour importation

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

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

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

Dans cet exemple, le pool `tank` est disponible pour être importé dans le système cible. Chaque pool est identifié par un nom et un identifiant numérique unique. Si plusieurs pools portant le

même nom sont disponibles pour l'import, vous pouvez utiliser leur identifiant numérique afin de les distinguer.

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

```
# zpool import
pool: tank
id: 4715259469716913940
state: DEGRADED
status: One or more devices are unavailable.
action: The pool can be imported despite missing or damaged devices.  The
fault tolerance of the pool may be compromised if imported.
config:

tank                        DEGRADED
mirror-0                    DEGRADED
c0t5000C500335E106Bd0       ONLINE
c0t5000C500335FC3E7d0       UNAVAIL  cannot open

device details:

c0t5000C500335FC3E7d0     UNAVAIL   cannot open
status: ZFS detected errors on this device.
The device was missing.
```

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

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

```
# zpool import
pool: mothership
id: 3702878663042245922
state: UNAVAIL
status: One or more devices are unavailable.
action: The pool cannot be imported due to unavailable devices or data.
config:

mothership     UNAVAIL  insufficient replicas
```

```
raidz1-0     UNAVAIL  insufficient replicas
c8t0d0     UNAVAIL  cannot open
c8t1d0     UNAVAIL  cannot open
c8t2d0     ONLINE
c8t3d0     ONLINE

device details:

c8t0d0    UNAVAIL          cannot open
status: ZFS detected errors on this device.
The device was missing.

c8t1d0    UNAVAIL          cannot open
status: ZFS detected errors on this device.
The device was missing.
```

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

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

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

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

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

Importation de pools de stockage ZFS

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

```
# zpool import tank
```

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

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

dozer       ONLINE
c1t9d0    ONLINE

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

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

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

```
# zpool import dozer zeepool
```

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

Remarque - Vous ne pouvez pas renommer un pool directement. Vous pouvez uniquement modifier le nom d'un pool pendant son exportation et son importation.

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

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

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

Les pools peuvent également être importés sous un root de remplacement à l'aide de l'option -R. Pour plus d'informations sur les pools root de remplacement, reportez-vous à “Utilisation d'un pool ZFS avec un emplacement de root de remplacement” à la page 279

Importation d'un pool avec un périphérique de journalisation manquant

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

```
# zpool import dozer
pool: dozer
id: 16216589278751424645
state: UNAVAIL
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://support.oracle.com/msg/ZFS-8000-6X
config:

dozer             UNAVAIL  missing device
mirror-0         ONLINE
c8t0d0  ONLINE
c8t1d0  ONLINE

device details:

missing-1        UNAVAIL          corrupted data
status: ZFS detected errors on this device.
The device has bad label or disk contents.


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

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

```
# zpool import -m dozer
# zpool status dozer
pool: dozer
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
Run 'zpool status -v' to see device specific details.
scan: none requested
```

```
config:

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

errors: No known data errors
```

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

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

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

cannot import 'dozer': one or more devices is currently unavailable
# zpool import -m dozer
# zpool status dozer
pool: dozer
state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: https://support.oracle.com/epmos/faces/KmHome?_adf.ctrl-
state=10oxbvnj5n_4&_afrLoop=1145647522713
scan: scrub repaired 0 in 0h0m with 0 errors on Fri Oct 15 16:51:39 2010
config:

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

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

Importation d'un pool en mode lecture seule

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

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

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

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

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

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

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

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

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

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

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

Récupération de pools de stockage ZFS détruits

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

```
# zpool destroy tank
# zpool import -D
pool: tank
id: 5154272182900538157
state: ONLINE (DESTROYED)
action: The pool can be imported using its name or numeric identifier.
config:

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

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

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

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

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

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

errors: No known data errors
```

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

```
# zpool destroy dozer
# zpool import -D
pool: dozer
id: 4107023015970708695
state: DEGRADED (DESTROYED)
status: One or more devices are unavailable.
action: The pool can be imported despite missing or damaged devices.  The
fault tolerance of the pool may be compromised if imported.
```

```
config:

dozer              DEGRADED
raidz2-0         DEGRADED
c8t0d0         ONLINE
c8t1d0         ONLINE
c8t2d0         ONLINE
c8t3d0         UNAVAIL  cannot open
c8t4d0         ONLINE

device details:

c8t3d0    UNAVAIL          cannot open
status: ZFS detected errors on this device.
The device was missing.
# zpool import -Df dozer
# zpool status -x
pool: dozer
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
Run 'zpool status -v' to see device specific details.
scan: none requested
config:

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

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

Mise à niveau de pools de stockage ZFS

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

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

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

```
# zpool upgrade -v
This system is currently running ZFS pool version 33.

The following versions are supported:

VER  DESCRIPTION
---  --------------------------------------------------------
1   Initial ZFS version
2   Ditto blocks (replicated metadata)
3   Hot spares and double parity RAID-Z
4   zpool history
5   Compression using the gzip algorithm
6   bootfs pool property
7   Separate intent log devices
8   Delegated administration
9   refquota and refreservation properties
10  Cache devices
11  Improved scrub performance
12  Snapshot properties
13  snapused property
14  passthrough-x aclinherit
15  user/group space accounting
16  stmf property support
17  Triple-parity RAID-Z
18  Snapshot user holds
19  Log device removal
20  Compression using zle (zero-length encoding)
21  Deduplication
22  Received properties
23  Slim ZIL
24  System attributes
25  Improved scrub stats
26  Improved snapshot deletion performance
27  Improved snapshot creation performance
28  Multiple vdev replacements
```

```
29  RAID-Z/mirror hybrid allocator
30  Encryption
31  Improved 'zfs list' performance
32  One MB blocksize
33  Improved share support
34  Sharing with inheritance

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

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

```
# zpool upgrade -a
```

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

♦ ♦ ♦ CHAPITRE 4

Gestion des composants du pool root ZFS

Ce chapitre décrit comment gérer les composants du pool root ZFS Oracle Solaris, par exemple pour connecter un miroir de pool root, cloner un environnement d'initialisation ZFS et redimensionner des périphériques de swap et de vidage.

Ce chapitre contient les sections suivantes :

Pour plus d'informations sur la récupération de pool root, reportez-vous à “ Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ”.

Pour toutes informations récentes importantes, reportez-vous au manuel accompagnant la version Oracle Solaris 11.2.

Gestion des composants du pool root ZFS

ZFS est le système de fichiers root par défaut de la dernière version d'Oracle Solaris 11. Passez en revue les considérations suivantes lorsque vous installez la version d'Oracle Solaris.

- **Installation** : la version permet d'installer et d'initialiser un système de fichiers root ZFS des manières suivantes :
  - Live CD (x86 uniquement) : installe un pool root ZFS sur un seul disque. Vous pouvez utiliser le menu de partition `fdisk` au cours de l'installation afin de partitionner le disque pour votre environnement.
  - Installation en mode texte (SPARC et x86) : installe un pool root ZFS sur un seul disque à partir d'un média ou sur le réseau. Vous pouvez utiliser le menu de partition `fdisk` au cours de l'installation afin de partitionner le disque pour votre environnement.
  - Programme d'installation automatisée (AI) (SPARC et x86) : installe automatiquement un pool root ZFS. Vous pouvez utiliser un fichier manifeste AI afin de déterminer le disque et les partitions de disque à utiliser pour le pool root ZFS.

- **Périphériques de swap et de vidage** : créés automatiquement sur les volumes ZFS dans le pool root ZFS par toutes les méthodes d'installation ci-dessus. Pour plus d'informations sur la gestion des périphériques de swap et de vidage, reportez-vous à la section “Gestion de vos périphériques de swap et de vidage ZFS” à la page 121.
- **Configuration d'un pool root mis en miroir** : vous pouvez configurer un pool root mis en miroir lors d'une installation automatique. Pour plus d'informations sur la configuration d'un pool root mis en miroir après une installation, reportez-vous à la section “Configuration d'un pool root mis en miroir (SPARC ou x86/VTOC)” à la page 111.
- **Gestion de l'espace de pool root** : après l'installation du système, envisagez de définir un quota sur le système de fichiers root ZFS pour empêcher qu'il ne se remplisse. A l'heure actuelle, aucun espace de pool root ZFS n'est réservé en tant que filet de sécurité pour un système de fichiers plein. Par exemple, si le disque pour le pool root est de 68 Go, définissez un quota de 67 Go sur le système de fichiers root ZFS (`rpool/ROOT/solaris`), ce qui permet au système de fichiers de conserver 1 Go d'espace. Pour plus d'informations sur la définition de quotas, reportez-vous à la section “Définitions de quotas sur les systèmes de fichiers ZFS” à la page 186.
- **Migration ou récupération de pool root** Considérez la création d'une archive de récupération de pool root pour la reprise sur sinistre ou à des fins de migration à l'aide de l'utilitaire d'archivage Oracle Solaris. Pour plus d'informations, reportez-vous à “ Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ” et à la dans la page de manuel `archiveadm`(1M).

Configuration requise pour le pool root ZFS

Consultez les sections ci-après décrivant l'espace de pool root ZFS et la configuration requise.

Espace de pool root ZFS requis

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

Consultez les exigences en termes d'espace de pool de stockage ZFS suivantes :

- Pour une description de la mémoire requise pour les différentes méthodes d'installation, reportez-vous à “ Oracle Solaris 11.2 Release Notes ”.
- 7-13 Go d'espace disque minimum sont recommandés. L'espace est utilisé comme suit :
  - **Zone de swap et périphérique de vidage** : les capacités par défaut des volumes de swap et de vidage créés par les programmes d'installation de Solaris varient en fonction de la quantité de mémoire disponible sur le système et d'autres variables. La taille du périphérique de vidage correspond à la moitié de la mémoire physique du système, ou plus en fonction de l'activité du système.

Vous pouvez ajuster librement les tailles respectives des volumes de swap et de vidage, dès lors que celles-ci permettent au programme de fonctionner correctement 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 122.

- **Environnement d'initialisation (BE)** : un environnement d'initialisation ZFS est d'environ 4 à 6 Go. Un environnement d'initialisation ZFS cloné à partir d'un autre environnement d'initialisation ZFS ne requiert pas d'espace disque supplémentaire. Tenez compte du fait que la taille de l'environnement d'initialisation augmente lorsqu'il est mis à jour, ce en fonction des mises à jour. Tous les environnements d'initialisation ZFS d'un même pool root utilisent les mêmes périphériques de swap et de vidage.
- **Composants du SE Oracle Solaris** : à l'exception de `/var`, tous les sous-répertoires du système de fichiers root qui font partie de l'image du SE doivent se trouver dans le système de fichiers root. En outre, tous les composants du SE Solaris doivent se trouver dans le pool root, à l'exception des périphériques de swap et de vidage.

Configuration requise pour le pool root ZFS

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

- Le disque destiné au pool root peut avoir une EFI (GPT) sur un système basé sur SPARC GPT prenant en charge les étiquettes de microprogramme ou avec un système x86, dans la plupart des cas. Toutefois, une étiquette SMI (VTOC) est appliquée sur un système SPARC sans microprogramme compatible GPT. Pour plus d'informations sur l'étiquette EFI (GPT), reportez-vous à “Utilisation de disques dans un pool de stockage ZFS” à la page 33.
- Le pool doit exister sur une ou plusieurs tranches de disque qui sont mises en miroir si le disque est étiqueté SMI (VTOC) existe. Si les disques de pool root sont étiquetés EFI (GPT), le pool peut exister sur un disque entier ou des disques entiers en miroir. Si vous tentez d'utiliser une configuration de pool non prise en charge lors d'une opération `beadm`, un message du type suivant s'affiche :

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

  Pour obtenir une description détaillée des configurations de pool root ZFS prises en charge, reportez-vous à la section “Création d'un pool root ZFS” à la page 43.
- Sur un système x86, le disque doit contenir une partition `fdisk` Solaris. 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 “ Utilisation de l’option fdisk ” du manuel “ Gestion des périphériques dans Oracle Solaris 11.2 ”.
- Les propriétés d'un pool ou les propriétés du système de fichiers peuvent être définies sur un pool root au cours d'une installation automatique. L'algorithme de compression `gzip` n'est pas pris en charge sur les pools root.
- Ne renommez pas le pool root une fois qu'il a été créé par une installation initiale. Si vous renommez le pool root, cela peut empêcher l'initialisation du système.

- N'utilisez pas de périphérique VMware avec allocation fine pour un périphérique de pool root.

Gestion de votre pool root ZFS

Les sections suivantes fournissent des informations sur l'installation et la mise à jour d'un pool root ZFS et la configuration d'un pool root en miroir.

Installation d'un pool root ZFS

Reportez - vous aux points suivants ZFS méthodes d'installation pour l'installation d'un pool root :

- La méthode d'installation de Live CD installe un pool root ZFS par défaut sur un seul disque.
- Le programme d'installation automatisée permet de disposer d'une certaine souplesse en installant un pool root ZFS sur le disque d'initialisation par défaut ou sur un disque cible que vous identifiez. Vous pouvez spécifier le périphérique logique, tel que `c1t0d0`, ou le chemin du périphérique physique. En outre, vous pouvez utiliser l'identificateur MPxIO ou l'ID du périphérique à installer.
- La méthode d'installation automatisée (AI, Automated Installation) vous permet de créer un manifeste AI pour identifier le disque ou les disques mis en miroir du pool root ZFS. Par exemple, les mots-clés suivants dans ce manifeste AI par défaut installe un pool root mis en miroir fragment de code de deux disques :

```
<target>
<disk whole_disk="true" in_zpool="rpool" in_vdev="mirrored">
<disk_name name="c1t0d0" name_type="ctd"/>
</disk>
<disk whole_disk="true" in_zpool="rpool" in_vdev="mirrored">
<disk_name name="c2t0d0" name_type="ctd"/>
</disk>
<logical>
<zpool name="rpool" is_root="true">
<vdev name="mirrored" redundancy="mirror"/>
<!--
```

Par exemple, si vous souhaitez définir des propriétés de pools dans le manifeste par défaut, il serait bon d'inclure les mots-clés `pool_options` après le système de fichiers mais avant les mots-clés `be_name`, comme dans l'exemple suivant :

```
 -->
```

```
<filesystem name="export" mountpoint="/export"/>
<filesystem name="export/home"/>
<pool_options>
<option name="listsnaps" value="on"/>
</pool_options>
<be name="solaris"/>
</zpool>
</logical>
</target>
```

Dans la syntaxe ci-dessus, la propriété de pool `listsnaps` est activée sur le pool root.

Après l'installation, examinez les informations de votre pool de stockage ZFS et du système de fichiers, qui peuvent varier selon le type d'installation et les personnalisations. Par exemple :

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

NAME        STATE     READ WRITE CKSUM
rpool       ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
c8t0d0  ONLINE       0     0     0
c8t1d0  ONLINE       0     0     0
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    11.8G  55.1G  4.58M  /rpool
rpool/ROOT               3.57G  55.1G    31K  legacy
rpool/ROOT/solaris       3.57G  55.1G  3.40G  /
rpool/ROOT/solaris/var    165M  55.1G   163M  /var
rpool/VARSHARE           42.5K  55.1G  42.5K  /var/share
rpool/dump               6.19G  55.3G  6.00G  -
rpool/export               63K  55.1G    32K  /export
rpool/export/home          31K  55.1G    31K  /export/home
rpool/swap               2.06G  55.2G  2.00G  -
```

Passez en revue les informations sur l'environnement d'initialisation ZFS. Par exemple :

```
# beadm list
BE      Active Mountpoint Space Policy Created
--      ------ ---------- ----- ------ -------
solaris NR     /          3.75G static 2012-07-20 12:10
```

Dans la sortie ci-dessus, le champ `Active` indique si l'environnement d'initialisation est actif maintenant (lettre `N`), actif lors de la réinitialisation (lettre `R`), ou les deux (lettres `NR`).

Mise à jour de l'environnement d'initialisation ZFS

L'environnement d'initialisation ZFS par défaut est nommé `solaris` par défaut. Vous pouvez identifier votre environnement d'initialisation en utilisant la commande `beadm list`. Par exemple :

```
# beadm list
BE      Active Mountpoint Space Policy Created
--      ------ ---------- ----- ------ -------
solaris NR     /          3.82G static 2012-07-19 13:44
```

Dans la sortie ci-dessus, NR signifie que l'environnement d'initialisation est actuellement actif et qu'il sera l'environnement d'initialisation actif après la réinitialisation.

La commande `pkg update` vous permet de mettre à jour votre environnement d'initialisation ZFS. Si vous mettez à jour votre environnement d'initialisation ZFS à l'aide de la commande `pkg update`, un nouvel environnement d'initialisation est créé et activé automatiquement, sauf si les mises à jour appliquées à l'environnement d'initialisation existant sont très minimes.

1. **Mettez à jour votre environnement d'initialisation ZFS.**

```
# pkg update


DOWNLOAD                                  PKGS       FILES    XFER (MB)
Completed                                707/707 10529/10529  194.9/194.9
.
.
.
```

Un nouvel environnement d'initialisation, `solaris-1`, est automatiquement créé et activé.

Vous pouvez également créer et activer un environnement d'initialisation de sauvegarde en dehors du processus de mise à jour.

```
# beadm create solaris-1
# beadm activate solaris-1
```

2. **Réinitialisez le système pour terminer l'activation de l'environnement d'initialisation. Ensuite, confirmez le statut de l'environnement d'initialisation.**

```
# init 6
.
.
.
# beadm list
BE        Active Mountpoint Space  Policy Created
--        ------ ---------- -----  ------ -------
solaris   -      -          46.95M static 2012-07-20 10:25
solaris-1 NR     /          3.82G  static 2012-07-19 14:45
```

3. **Si une erreur se produit lors de l'initialisation du nouvel environnement d'initialisation, activez et initialisez sur l'environnement d'initialisation précédent.**

```
# beadm activate solaris
# init 6
```

Montage d'un environnement d'initialisation alternatif

A des fins de récupération, vous pouvez être amené à copier ou à accéder à un fichier à partir d'un autre environnement d'initialisation.

1. **Connectez-vous en tant qu'administrateur.**

2. **Montez l'environnement d'initialisation alternatif.**

```
# beadm mount solaris-1 /mnt
```

3. **Accédez à l'environnement d'initialisation.**

```
# ls /mnt
bin         export      media       pkg         rpool       tmp
boot        home        mine        platform    sbin        usr
dev         import      mnt         proc        scde        var
devices     java        net         project     shared
doe         kernel      nfs4        re          src
etc         lib         opt         root        system
```

4. **Démontez l'environnement d'initialisation alternatif lorsque vous avez terminé de l'utiliser.**

```
# beadm umount solaris-1
```

Configuration d'un pool root mis en miroir (SPARC ou x86/VTOC)

Si vous ne configurez pas de pool root mis en miroir au cours d'une installation automatique, vous pouvez facilement configurer un pool root mis en miroir après l'installation.

Pour plus d'informations sur le remplacement d'un disque dans un pool root, reportez-vous à la section “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/VTOC)” à la page 115.

1. **Affichez l'état du pool root actuel.**

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

NAME        STATE     READ WRITE CKSUM
rpool       ONLINE       0     0     0
c2t0d0s0    ONLINE       0     0     0

errors: No known data errors
```

2. **Préparez un second disque à raccorder au pool root, si nécessaire.**

   - SPARC : confirmez que le disque dispose d'une étiquette de disque SMI (VTOC) et d'une tranche 0. Si vous devez réétiqueter le disque et créer une tranche 0, reportez-vous à la section " Remplacement d'un pool root ZFS (VTOC) " du manuel " Gestion des périphériques dans Oracle Solaris 11.2 ".
   - x86 : confirmez que le disque dispose d'une partition `fdisk`, d'une étiquette de disque SMI et d'une tranche 0. Si vous devez repartitionner le disque et créer une tranche 0, reportez-vous à " Tranches intelligentes ou la modification des partitions " du manuel " Gestion des périphériques dans Oracle Solaris 11.2 ".

3. **Connectez un deuxième disque pour configurer un pool root mis en miroir.**

```
# zpool attach rpool c2t0d0s0 c2t1d0s0
Make sure to wait until resilver is done before rebooting.
```

   Le bon étiquetage et les blocs d'initialisation sont appliqués par défaut.

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

```
# zpool status rpool
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Fri Jul 20 13:39:53 2012
938M scanned
938M resilvered at 46.9M/s, 7.86% done, 0h3m to go
config:

NAME      STATE      READ  WRITE  CKSUM
rpool     DEGRADED      0      0      0
mirror-0  DEGRADED      0      0      0
c2t0d0s0  ONLINE        0      0      0
c2t1d0s0  DEGRADED      0      0      0  (resilvering)
```

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 :

```
resilvered 11.6G in 0h5m with 0 errors on Fri Jul 20 13:57:25 2012
```

5. **Si vous connectez un disque plus grand, définissez la propriété `autoexpand` du pool pour étendre la taille de pool.**

   Déterminez la taille du pool `rpool` existant :

```
# zpool list rpool
NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool  29.8G   152K  29.7G   0%  1.00x  ONLINE  -

# zpool set autoexpand=on rpool
```

   Vérifiez la taille du pool `rpool` étendu :

```
# zpool list rpool
NAME   SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool  279G   146K  279G   0%  1.00x  ONLINE  -
```

6. **Assurez-vous que vous pouvez initialiser correctement à partir du nouveau disque.**

▼ Configuration d'un pool root mis en miroir (x86/EFI (GPT))

La version Oracle Solaris 11.1 installe une étiquette EFI (GPT) par défaut sur un système x86 dans la plupart des cas.

Si vous ne configurez pas de pool root mis en miroir au cours d'une installation automatique, vous pouvez facilement configurer un pool root mis en miroir après l'installation.

Pour plus d'informations sur le remplacement d'un disque dans un pool root, reportez-vous à la section “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/VTOC)” à la page 115.

1. **Affichez l'état du pool root actuel.**

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

NAME     STATE     READ WRITE CKSUM
```

```
rpool    ONLINE       0     0     0
c2t0d0   ONLINE       0     0     0

errors: No known data errors
```

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

```
# zpool attach rpool c2t0d0 c2t1d0
Make sure to wait until resilver is done before rebooting.
```

Le bon étiquetage et les blocs d'initialisation sont appliqués par défaut.

Si vous avez personnalisé des partitions sur votre disque de pool root, vous pouvez avoir besoin d'une syntaxe similaire à la suivante :

```
# zpool attach rpool c2t0d0s0 c2t1d0
```

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

```
# zpool status rpool
pool: rpool
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
continue to function in a degraded state.
action: Wait for the resilver to complete.
Run 'zpool status -v' to see device specific details.
scan: resilver in progress since Fri Jul 20 13:52:05 2012
809M scanned
776M resilvered at 44.9M/s, 6.82% done, 0h4m to go
config:

NAME      STATE     READ WRITE CKSUM
rpool     DEGRADED     0     0     0
mirror-0  DEGRADED     0     0     0
c8t0d0    ONLINE       0     0     0
c8t1d0    DEGRADED     0     0     0  (resilvering)

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 :

```
resilvered 11.6G in 0h5m with 0 errors on Fri Jul 20 13:57:25 2012
```

4. Si vous connectez un disque plus grand, définissez la propriété `autoexpand` du pool pour étendre la taille de pool.

Déterminez la taille du pool `rpool` existant :

```
# zpool list rpool
NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool  29.8G   152K  29.7G   0%  1.00x  ONLINE  -

# zpool set autoexpand=on rpool
```

Vérifiez la taille du pool `rpool` étendu :

```
# zpool list rpool
NAME   SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool  279G   146K  279G   0%  1.00x  ONLINE  -
```

5. **Assurez-vous que vous pouvez initialiser correctement à partir du nouveau disque.**

▼ Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/VTOC)

Vous pouvez être amené à remplacer un disque dans le pool root pour les raisons suivantes :

- Le pool root est trop petit et vous souhaitez le remplacer par un disque plus grand.
- Le disque du pool root est défectueux. Dans un pool non redondant, si le disque est défectueux et empêche l'initialisation du système, vous devez initialiser votre système à partir d'un autre média, par exemple un CD ou le réseau, avant de remplacer le disque du pool root.
- Si vous exécutez la commande `zpool replace` pour remplacer un disque dans un disque de pool root, vous devrez appliquer les blocs d'initialisation manuellement.

Dans une configuration de pool root en miroir, vous pouvez peut-être tenter un remplacement de disque sans avoir à initialiser à partir d'un autre média. Vous pouvez remplacer un disque défaillant en utilisant la commande `zpool replace` ou, si vous avez un disque supplémentaire, la commande `zpool attach`. Pour savoir comment connecter un autre disque et déconnecter un disque de pool root, reportez-vous aux étapes ci-dessous.

Sur les systèmes équipés de disques SATA, vous devez déconnecter le disque et en supprimer la configuration avant de tenter d'utiliser la commande `zpool replace` pour remplacer un disque défectueux. Par exemple :

```
# zpool offline rpool c1t0d0s0
# cfgadm -c unconfigure c1::dsk/c1t0d0
<Physically remove failed disk c1t0d0>
<Physically insert replacement disk c1t0d0>
# cfgadm -c configure c1::dsk/c1t0d0
<Confirm that the new disk has an SMI label and a slice 0>
# zpool replace rpool c1t0d0s0
# zpool online rpool c1t0d0s0
# zpool status rpool
<Let disk resilver before installing the boot blocks>
# bootadm install-bootloader
```

Avec certains composants matériels, il n'est pas nécessaire de connecter le disque, ni de reconfigurer son remplacement après son insertion.

1. **Connectez physiquement le disque de remplacement.**

2. **Préparez un second disque à raccorder au pool root, si nécessaire.**

   - SPARC : confirmez que le (nouveau) disque de remplacement dispose d'une étiquette SMI (VTOC) et d'une tranche 0. Pour plus d'informations sur le réétiquetage d'un disque destiné au pool root, reportez-vous à la section " Etiquetage d'un disque " du manuel " Gestion des périphériques dans Oracle Solaris 11.2 ".
   - x86 : confirmez que le disque dispose d'une partition `fdisk`, d'une étiquette de disque SMI et d'une tranche 0. Si vous devez repartitionner le disque et créer une tranche 0, reportez-vous aux sections relatives aux étiquettes et aux partitions dans " Configuration des disques " du manuel " Gestion des périphériques dans Oracle Solaris 11.2 ".

3. **Associez le nouveau disque au pool root.**

   Par exemple :

   ```
   # zpool attach rpool c2t0d0s0 c2t1d0s0
   Make sure to wait until resilver is done before rebooting.
   ```

   Le bon étiquetage et les blocs d'initialisation sont appliqués par défaut.

4. **Confirmez le statut du pool root.**

   Par exemple :

   ```
   # zpool status rpool
   pool: rpool
   state: ONLINE
   scan: resilvered 11.7G in 0h5m with 0 errors on Fri Jul 20 13:45:37 2012
   config:

   NAME      STATE     READ WRITE CKSUM
   rpool     ONLINE       0     0     0
   mirror-0  ONLINE       0     0     0
   c2t0d0s0  ONLINE       0     0     0
   c2t1d0s0  ONLINE       0     0     0

   errors: No known data errors
   ```

5. **Vérifiez que vous pouvez initialiser à partir du nouveau disque une fois la réargenture terminée.**

   Par exemple, sur un système SPARC :

   ```
   ok boot /pci@1f,700000/scsi@2/disk@1,0
   ```

   Identifiez 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 suivant, le disque du pool root actuel (`c2t0d0s0`) est :

   ```
   /pci@1f,700000/scsi@2/disk@0,0
   ```

Dans l'exemple suivant, le disque d'initialisation de remplacement est (`c2t1d0s0`) :

```
boot /pci@1f,700000/scsi@2/disk@1,0
```

6. **Si le système s'initialise à partir du nouveau disque, déconnectez l'ancien disque.**

   Par exemple :

   ```
   # zpool detach rpool c2t0d0s0
   ```

7. **Si vous remplacez un disque de pool root par un disque plus grand, définissez la propriété de pool `autoexpand` pour augmenter la taille du pool.**

   Déterminez la taille du pool `rpool` existant :

   ```
   # zpool list rpool
   NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
   rpool  29.8G   152K  29.7G   0%  1.00x  ONLINE  -

   # zpool set autoexpand=on rpool
   ```

   Vérifiez la taille du pool `rpool` étendu :

   ```
   # zpool list rpool
   NAME   SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
   rpool  279G   146K  279G   0%  1.00x  ONLINE  -
   ```

8. **Configurez le système de manière à ce qu'il s'initialise automatiquement à partir du nouveau disque.**

   - SPARC : configurez le système de manière à ce qu'il s'initialise automatiquement à partir du nouveau disque, soit en utilisant la commande `eeprom`, soit en utilisant la commande `setenv` de la PROM d'initialisation.
   - x86 : reconfigurez le BIOS du système.

▼ Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/EFI (GPT))

Vous pouvez être amené à remplacer un disque dans le pool root pour les raisons suivantes :

- Le pool root est trop petit et vous souhaitez le remplacer par un disque plus grand.
- Le disque du pool root est défectueux. Dans un pool non redondant, si le disque est défectueux et empêche l'initialisation du système, vous devez initialiser votre système à partir d'un autre média, par exemple un CD ou le réseau, avant de remplacer le disque du pool root.

- Si vous exécutez la commande `zpool replace` pour remplacer un disque dans un disque de pool root, vous devrez appliquer les blocs d'initialisation manuellement.

Dans une configuration de pool root en miroir, vous pouvez peut-être tenter un remplacement de disque sans avoir à initialiser à partir d'un autre média. Vous pouvez remplacer un disque défaillant en utilisant la commande `zpool replace` ou, si vous avez un disque supplémentaire, la commande `zpool attach`. Pour savoir comment connecter un autre disque et déconnecter un disque de pool root, reportez-vous aux étapes ci-dessous.

Sur les systèmes équipés de disques SATA, vous devez déconnecter le disque et en supprimer la configuration avant de tenter d'utiliser la commande `zpool replace` pour remplacer un disque défectueux. Par exemple :

```
# zpool offline rpool c1t0d0
# cfgadm -c unconfigure c1::dsk/c1t0d0
<Physically remove failed disk c1t0d0>
<Physically insert replacement disk c1t0d0>
# cfgadm -c configure c1::dsk/c1t0d0
# zpool online rpool c1t0d0
# zpool replace rpool c1t0d0
# zpool status rpool
<Let disk resilver before installing the boot blocks>
x86# bootadm install-bootloader
```

Avec certains composants matériels, il n'est pas nécessaire de connecter le disque, ni de reconfigurer son remplacement après son insertion.

1. **Connectez physiquement le disque de remplacement.**

2. **Associez le nouveau disque au pool root.**

   Par exemple :

   ```
   # zpool attach rpool c2t0d0 c2t1d0
   Make sure to wait until resilver is done before rebooting.
   ```

   Le bon étiquetage et les blocs d'initialisation sont appliqués par défaut.

3. **Confirmez le statut du pool root.**

   Par exemple :

   ```
   # zpool status rpool
   pool: rpool
   state: ONLINE
   scan: resilvered 11.6G in 0h5m with 0 errors on Fri Jul 20 12:06:07 2012
   config:

   NAME      STATE     READ WRITE CKSUM
   rpool     ONLINE       0     0     0
   mirror-0  ONLINE       0     0     0
   c2t0d0    ONLINE       0     0     0
   ```

```
c2t1d0    ONLINE       0     0     0

errors: No known data errors
```

4. **Vérifiez que vous pouvez initialiser à partir du nouveau disque une fois la réargenture terminée.**

5. **Si le système s'initialise à partir du nouveau disque, déconnectez l'ancien disque.**

   Par exemple :

   ```
   # zpool detach rpool c2t0d0
   ```

6. **Si vous remplacez un disque de pool root par un disque plus grand, définissez la propriété de pool `autoexpand` pour augmenter la taille du pool.**

   Déterminez la taille du pool `rpool` existant :

   ```
   # zpool list rpool
   NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
   rpool  29.8G   152K  29.7G   0%  1.00x  ONLINE  -

   # zpool set autoexpand=on rpool
   ```

   Vérifiez la taille du pool `rpool` étendu :

   ```
   # zpool list rpool
   NAME   SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
   rpool  279G   146K  279G   0%  1.00x  ONLINE  -
   ```

7. **Configurez le système de manière à ce qu'il s'initialise automatiquement à partir du nouveau disque.**

   Reconfigurez le BIOS du système.

▼ Création d'un environnement d'initialisation dans un pool root différent (SPARC ou x86/EFI (GPT))

Si vous souhaitez recréer votre environnement d'initialisation existant dans un autre pool root, effectuez les étapes décrites dans cette procédure. Vous pouvez modifier les étapes en fonction de ce que vous souhaitez obtenir :deux pools root dotés d'environnements d'initialisation similaires ayant des périphériques de swap et de vidage indépendants ou un environnement d'initialisation dans un autre pool root qui partage les périphériques de swap et de vidage.

Une fois que vous avez activé et initialisé à partir du nouvel environnement d'initialisation dans le second pool root, celui-ci ne disposera d'aucune information sur l'environnement d'initialisation précédent du premier pool root. Si vous souhaitez revenir à l'environnement

d'initialisation d'origine, réinitialisez le système manuellement à partir du disque d'initialisation du pool root d'origine.

1. **Créez le pool root de remplacement.**

   ```
   # zpool create -B rpool2 c2t2d0
   ```

   Sinon, créez un pool root de remplacement mis en miroir. Par exemple :

   ```
   # zpool create -B rpool2 mirror c2t2d0 c2t3d0
   ```

2. **Créez le nouvel environnement d'initialisation dans le deuxième pool root. Par exemple :**

   ```
   # beadm create -p rpool2 solaris2
   ```

3. **Appliquez les informations d'initialisation au second pool root. Par exemple :**

   ```
   # bootadm install-bootloader -P rpool2
   ```

4. **Définissez la propriété `bootfs` sur le deuxième pool root. Par exemple :**

   ```
   # zpool set bootfs=rpool2/ROOT/solaris2 rpool2
   ```

5. **Activez le nouvel environnement d'initialisation. Par exemple :**

   ```
   # beadm activate solaris2
   ```

6. **Initialisez à partir du nouvel environnement d'initialisation.**

   - SPARC : configurez le système de manière à ce qu'il s'initialise automatiquement à partir du nouveau disque, soit en utilisant la commande `eeprom`, soit en utilisant la commande `setenv` de la PROM d'initialisation.
   - x86 : reconfigurez le BIOS du système.

   Votre système doit s'exécuter sous le nouvel environnement d'initialisation.

7. **Recréez le volume de swap. Par exemple :**

   ```
   # zfs create -V 4g rpool2/swap
   ```

8. **Mettez à jour l'entrée `/etc/vfstab` pour le nouveau périphérique de swap. Par exemple :**

   ```
   /dev/zvol/dsk/rpool2/swap       -               -               swap -     no      -
   ```

9. **Recréez le volume de vidage. Par exemple :**

   ```
   # zfs create -V 4g rpool2/dump
   ```

10. **Réinitialisez le périphérique de vidage. Par exemple :**

```
# dumpadm -d /dev/zvol/dsk/rpool2/dump
```

11. **Réinitialisez pour effacer les périphériques de swap et de vidage du pool root d'origine.**

```
# init 6
```

Gestion de vos périphériques de swap et de vidage ZFS

Au cours du processus d'installation, une zone de swap est créée sur un volume ZFS du pool root ZFS. Par exemple :

```
# swap -l
swapfile                  dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap 145,2        16 16646128 16646128
```

Au cours du processus d'installation, un périphérique de vidage est créé sur un volume ZFS du pool root ZFS. En règle générale, un périphérique de vidage ne nécessite aucune administration car il est créé automatiquement lors de l'installation. Par exemple :

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/
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 la taille des volumes de swap et de vidage créés par les programmes d'installation, reportez-vous à la section “Configuration requise pour le pool root ZFS” à la page 106.

La taille des volumes de swap et de vidage peut être ajustée 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 122.

Considérez les points suivants lorsque vous travaillez avec des périphériques de swap et de vidage ZFS :

- Si vous souhaitez créer des périphériques de swap et de vidage dans un pool non-root, ne créez pas des volumes de swap et de vidage dans un pool RAID-Z. Si un pool inclut les volumes de swap et de vidage, il doit s'agir d'un pool à un seul disque ou d'un pool mis en miroir. Sinon, un message similaire à celui figurant ci-dessous s'affiche :

  ```
  /dev/zvol/dsk/rzpool/swap: Operation not supported
  ```

- Créer un volume de swap ou de vidage dans un pool non root. L'exécution de la commande `dumpadm -d` met en évidence la réinitialisation du périphérique de vidage.

```
# zfs create -V 10g bpool/dump2
# dumpadm -d /dev/zvol/dsk/bpool/dump2
Dump content      : kernel with ZFS metadata
Dump device       : /dev/zvol/dsk/bpool/dump2 (dedicated)
Savecore directory: /var/crash
Savecore enabled  : yes
Save compressed   : on
```

- Vous devez utiliser des volumes ZFS distincts pour les périphériques de swap et de vidage.
- Volumes ne sont pas prises en charge pour des volumes de swap.
- L'utilisation d'un fichier swap sur un système de fichiers ZFS n'est actuellement pas prise en charge.
- Si vous devez modifier votre zone de swap ou votre périphérique de vidage après l'installation du système, utilisez les commandes `swap` et `dumpadm` de la même manière que dans les versions précédentes de Solaris. Pour plus d'informations, reportez-vous au Chapitre 3, “ Extension de l’espace de swap ” du manuel “ Gestion des systèmes de fichiers dans Oracle Solaris 11.2 ” et “ Dépannage des problèmes d’administration système dans Oracle Solaris 11.2 ”.

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

Il peut s'avérer nécessaire d'ajuster la taille des périphériques de swap et de vidage ou encore de recréer les volumes de swap et de vidage.

- Vous pouvez rétablir la propriété `volsize` du périphérique de vidage après l'installation d'un système. Par exemple :

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

- Vous pouvez redimensionner le volume de swap et les utiliser immédiatement par le système. Par exemple :

```
# swap -l
swapfile                     dev    swaplo    blocks     free
/dev/zvol/dsk/rpool/swap   303,1         8   2097144  2097144
# zfs get volsize rpool/swap
NAME         PROPERTY   VALUE    SOURCE
rpool/swap   volsize       1G     local
```

```
# zfs set volsize=2g rpool/swap
# swap -l
swapfile                     dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap   303,1         8  2097144  2097144
/dev/zvol/dsk/rpool/swap   303,1   2097160  2097144  2097144
```

Vous pouvez également utiliser la méthode suivante pour redimensionner le volume d'échange. Dans le cadre de cette méthode, mais dans ce cas, vous devez réinitialiser le système pour constater la modification.

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs set volsize=2G rpool/swap
# swap -a /dev/zvol/dsk/rpool/swap
# init 6
```

Remarque - Par défaut, lorsque vous spécifiez *n* blocs pour la taille de la zone de swap, la première page du fichier de swap est automatiquement ignorée. Par conséquent, la taille réelle, qui est affectée est *n*-1 blocs. Pour configurer la taille du fichier swap `swaplow` différemment, utilisez l'option à l'aide de la commande `swap`. Pour plus d'informations sur les options de la commande `swap`, reportez-vous à la page de manuel swap(1M).

Pour plus d'informations sur la suppression d'un périphérique de swap sur un système actif, reportez-vous à“ Ajout d’espace de swap dans un environnement root ZFS Oracle Solaris ” du manuel “ Gestion des systèmes de fichiers dans Oracle Solaris 11.2 ”.

- Si vous avez besoin de plus d'espace de swap sur un système déjà installé et le périphérique de swap est occupé, il suffit d'ajouter un autre volume de swap. Par exemple :

  ```
  # zfs create -V 2G rpool/swap2
  ```

- Activez le nouveau volume de swap. Par exemple :

  ```
  # swap -a /dev/zvol/dsk/rpool/swap2
  # swap -l
  swapfile                  dev  swaplo   blocks   free
  /dev/zvol/dsk/rpool/swap  256,1      16 1058800 1058800
  /dev/zvol/dsk/rpool/swap2 256,3      16 4194288 4194288
  ```

- Ajoutez une entrée pour le deuxième volume de swap dans le fichier `/etc/vfstab`. Par exemple :

  ```
  /dev/zvol/dsk/rpool/swap2     -            -            swap    -    no    -
  ```

Dépannage du périphérique de vidage ZFS

Vérifiez les éléments suivants si vous rencontrez des problèmes soit lors de la capture d'un vidage sur incident du système, soit lors du redimensionnement du périphérique de vidage.

- Si un vidage sur incident n'a pas été automatiquement créé, vous pouvez utiliser la commande `savecore` pour enregistrer le vidage sur incident.
- Lorsque vous installez un système de fichiers root ZFS ou lorsque vous effectuez une migration vers un système de fichiers root ZFS pour la première fois, un périphérique de vidage est automatiquement créé. Dans la plupart des cas, vous devez uniquement ajuster la taille par défaut du périphérique de vidage si celle-ci est trop petite. Par exemple, vous pouvez augmenter la taille du périphérique 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 périphérique de vidage de grande capacité 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/
  Savecore enabled: yes
  Save compressed: on
  ```

- 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 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 le dimensionnement des périphériques de swap et de vidage, reportez-vous à“ Planification de l’espace de swap ” du manuel “ Gestion des systèmes de fichiers dans Oracle Solaris 11.2 ”.
- 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 root. Ce dernier ne peut pas contenir plusieurs périphériques de niveau supérieur.

Initialisation à partir d'un système de fichiers root ZFS

Les systèmes SPARC et les systèmes x86 s'initialisent à l'aide d'une archive d'initialisation, qui est une image de système de fichiers contenant les fichiers requis pour l'initialisation. Si vous procédez à l'initialisation à partir d'un système de fichiers root ZFS, le chemin d'accès à l'archive et le nom du fichier de noyau sont résolus dans le système de fichiers root (jeu de données) sélectionné pour l'initialisation.

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

Par défaut, le système de fichiers sélectionné pour l'initialisation est celui qui est identifié par la propriété `bootfs` du pool. Il est possible de passer outre à cette sélection par défaut en spécifiant un autre système de fichiers amorçable inclus dans la commande `boot -Z` sur un système SPARC ou en sélectionnant un autre périphérique d'initialisation à partir du BIOS sur un système x86.

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

Tenez compte des points suivants lors de l'initialisation à partir d'un disque de pool root ZFS mis en miroir :

- Vous pouvez connecter un disque pour créer un pool root ZFS en miroir après l'installation. Pour plus d'informations sur la création d'un pool root mis en miroir, reportez-vous à la section “Configuration d'un pool root mis en miroir (SPARC ou x86/VTOC)” à la page 111.
- En ligne conserver vos et il est joint à - vis des disques de pool root afin qu'il puisse s'initialiser à partir de n'importe lequel d'entre eux, si nécessaire.
- Vous ne pouvez pas initialiser le système directement à partir d'un disque dans le système a été détachée à l'aide de la commande `zpool detach`. Vous ne pouvez pas non plus s'initialiser à partir d'un disque de pool root qui active est actuellement hors ligne. Cependant, sur un système x86 avec une architecture et le numéro d'ordre d'initialisation BIOS est défini correctement et si le pool root, le système est mis en miroir à partir du

second disque d'initialisation même si la base de données principale automatiquement hors ligne ou le disque d'initialisation détachée.

- **SPARC** : Le disque principal dans un pool root en miroir est généralement l'unité d'initialisation par défaut. Vous pouvez initialiser à partir d'un autre périphérique dans un pool root ZFS mis en miroir, mais vous devrez démarrer le système à partir du disque de façon plus précise. Si vous souhaitez initialiser sur le solde pour les périphériques de pools root ou si vous souhaitez que à ce qu'il s'initialise automatiquement à partir du disque de pool root restant, vous devez mettre à jour le PROM unité d'initialisation pour indiquer que par défaut.

  Vous pouvez par exemple effectuer l'initialisation à partir de l'un des deux disques (`c1t0d0s0` ou `c1t1d0s0`) de ce pool.

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

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

  Indiquez le disque alternatif à l'invite `ok`.

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

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

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

- **x86** : Sur un système x86 avec une architecture de commande et le disque d'initialisation du BIOS définis correctement, le système s'initialise automatiquement à l'aide de la deuxième périphérique si le disque de pool root principal a été détaché, est indisponible ou hors ligne.

  Unité d'initialisation confirmer de l'environnement d'initialisation actif. Par exemple :

  ```
  x86# prtconf -v|sed -n '/bootpath/,/value/p'
  name='bootpath' type=string items=1
  value='/pci@0,0/pci8086,25f8@4/pci108e,286@0/disk@0,0:a'
  ```

- **SPARC or x86** : si vous remplacez un disque de pool root à l'aide de la commande `zpool replace`, vous devez installer les informations d'initialisation sur le nouveau disque à l'aide de la commande `bootadm`. Si vous créez un pool root ZFS mis en miroir à l'aide de la méthode d'installation initiale ou si vous utilisez la commande `zpool attach` pour connecter un disque au pool root, cette étape n'est pas nécessaire. La syntaxe `bootadm` est :

```
# bootadm install-bootloader
```

Si vous voulez installer le programme d'amorçage sur un autre pool root, utilisez l'option `-P` (pool).

```
# bootadm install-bootloader -P rpool2
```

Si vous voulez installer le programme d'amorçage hérité GRUB, utilisez la commande héritée `installgrub`.

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

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

Sur un système SPARC avec environnements d'initialisation ZFS multiples, vous pouvez initialiser à partir de tout environnement d'initialisation en utilisant la commande `beadm activate`.

Au cours de l'installation et de la procédure d'activation `beadm`, le système de fichiers root ZFS est automatiquement désigné avec la propriété `bootfs`.

Un pool peut contenir plusieurs systèmes de fichiers amorçables. Par défaut, l'entrée du système de fichiers d'initialisation dans le fichier */pool-name*`/boot/menu.lst` est identifiée par la propriété `bootfs` du pool. Cependant, une entrée `menu.lst` peut contenir une commande `bootfs` spécifiant un autre système de fichiers du pool. Le fichier `menu.lst` peut ainsi contenir les entrées de plusieurs systèmes de fichiers root du pool.

Lorsqu'un système est installé à l'aide d'un système de fichiers root ZFS, une entrée similaire à l'entrée suivante est ajoutée au fichier `menu.lst` :

```
title Oracle Solaris 11.2 SPARC
bootfs rpool/ROOT/solaris
```

Lorsqu'un nouvel environnement d'initialisation est créé, le fichier `menu.lst` est mis à jour automatiquement.

```
title Oracle Solaris 11.2 SPARC
bootfs rpool/ROOT/solaris
title solaris
bootfs rpool/ROOT/solaris2
```

Sur les systèmes SPARC, vous pouvez sélectionner l'environnement d'initialisation comme suit :

- Une fois qu'un environnement d'initialisation ZFS a été activé, vous pouvez utiliser la commande d'initialisation `-L` pour afficher la liste des systèmes de fichiers amorçables

contenus dans un pool ZFS. Vous pouvez ensuite sélectionner l'un des systèmes de fichiers amorçables de la liste. Des instructions détaillées concernant l'initialisation de ce système de fichiers s'affichent. Vous pouvez initialiser le système de fichiers sélectionné en suivant ces instructions.

- Utilisez la commande `-Z` *file system* pour initialiser un système de fichiers ZFS spécifique.

Cette méthode d'initialisation n'active pas l'environnement d'initialisation automatiquement. Une fois l'environnement d'initialisation initialisé avec la syntaxe `-L` et `-Z`, vous devez l'activer pour continuer à initialiser automatiquement depuis ce dernier.

EXEMPLE 4-1 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 de votre paramétrage lors de leur unité d'initialisation, vous pouvez utiliser la commande `beadm activate` pour indiquer la valeur BE par défaut.

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

```
# beadm list
BE        Active Mountpoint Space Policy Created
--        ------ ---------- ----- ------ -------
solaris   NR     /          3.80G static 2012-07-20 10:25
solaris-2 -      -          7.68M static 2012-07-19 13:44
```

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

```
ok boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a  File and args: -L
1 Oracle Solaris 11.2 SPARC
2 solaris
Select environment to boot: [ 1 - 2 ]: 1

To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/solaris-2

Program terminated
ok boot -Z rpool/ROOT/solaris-2
```

Gardez à l'esprit que l'environnement d'initialisation initialisé à l'aide de la commande ci-dessus n'est pas activé lors de la prochaine réinitialisation. Si vous souhaitez initialiser automatiquement à partir de l'environnement d'initialisation sélectionné lors de l'opération `boot -Z`, vous devez l'activer.

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

Dans Oracle Solaris 11, un système x86 est installé avec le GRUB hérité, les entrées suivantes sont ajoutées au fichier */pool-name* `/boot/grub/menu.lst` pendant le processus d'installation ou l'opération `beadm activate` pour initialiser ZFS automatiquement :

```
title solaris
bootfs rpool/ROOT/solaris
kernel$ /platform/i86pc/kernel/amd64/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/amd64/boot_archive
title solaris-1
bootfs rpool/ROOT/solaris-1
kernel$ /platform/i86pc/kernel/amd64/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/amd64/boot_archive
```

Si le périphérique identifié par GRUB comme périphérique d'initialisation contient un pool de stockage ZFS, le fichier `menu.lst` est utilisé pour créer le menu GRUB.

Sur un système x86 contenant plusieurs environnements d'initialisation ZFS, vous pouvez sélectionner un environnement d'initialisation à partir du menu GRUB. Si le système de fichiers root correspondant à cette entrée de menu est un système de fichiers ZFS, l'option suivante est ajoutée.

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

Dans Oracle Solaris 11.1, un système x86 est installé avec GRUB2. Le fichier `menu.lst` est remplacé par le fichier `/rpool/boot/grub/grub.cfg` , mais ce fichier ne doit pas être modifié manuellement. Utilisez la sous-commande `bootadm` pour ajouter, modifier et supprimer des entrées de menu.

Pour plus d'informations sur la modification des éléments du menu GRUB, reportez-vous au manuel “ Initialisation et arrêt des systèmes Oracle Solaris 11.2 ”.

EXEMPLE 4-2 x86 : Initialisation d'un système de fichiers ZFS

Lors de l'initialisation à partir d'un système de fichiers root ZFS sur un système GRUB2, le périphérique root est spécifié comme suit :

```
# bootadm list-menu
the location of the boot loader configuration files is: /rpool/boot/grub
default 0
console text
timeout 30
0 Oracle Solaris 11.2
```

Lors de l'initialisation à partir d'un système de fichiers root ZFS sur un système GRUB hérité, le périphérique root est spécifié par le paramètre d'initialisation `-B $ZFS-BOOTFS`. Par exemple :

```
title solaris
bootfs rpool/ROOT/solaris
kernel$ /platform/i86pc/kernel/amd64/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/amd64/boot_archive
title solaris-1
bootfs rpool/ROOT/solaris-1
kernel$ /platform/i86pc/kernel/amd64/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/amd64/boot_archive
```

EXEMPLE 4-3 x86 : réinitialisation rapide d'un système de fichiers root ZFS

La fonctionnalité de réinitialisation rapide offre la possibilité de réinitialiser en quelques secondes sur les systèmes x86. La fonctionnalité de réinitialisation rapide vous permet de réinitialiser vers un nouveau noyau sans subir les longs délais imposés par le BIOS et le programme d'amorçage. La possibilité de réinitialiser rapidement un système réduit considérablement les indisponibilités, tout en améliorant l'efficacité.

Vous devez continuer à utiliser la commande `init 6` lors du passage d'un environnement d'initialisation à un autre à l'aide de la commande `beadm activate`. Pour d'autres opérations système où la commande `reboot` est nécessaire, vous pouvez utiliser la commande `reboot -f`. Par exemple :

```
# reboot -f
```

Initialisation à des fins de récupération dans un environnement root ZFS

Suivez la procédure suivante si vous devez initialiser le système pour pouvoir récupérer un mot de passe root perdu ou tout problème similaire.

▼ Initialisation d'un système à des fins de récupération

La procédure ci-dessous vous permet de résoudre un problème lié à un problème de déclencher d'initialisation corrompu ou à un mot de passe root. Si vous devez remplacer un disque dans un pool root, reportez-vous à la section “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/VTOC)” à la page 115. Si vous avez besoin d'effectuer une récupération complète du système (à chaud), reportez-vous “ Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ”.

1. **Sélectionnez la méthode d'initialisation appropriée :**
   - x86 - Live Media : initialisez le système à partir du média d'installation et utilisez un terminal GNOME pour la procédure de récupération.
   - SPARC - Installation en mode texte : initialisez le système à partir du média d'installation ou du réseau, puis sélectionnez l'option `3 Shell` dans l'écran d'installation en mode texte.

- x86 - Installation en mode texte : dans le menu GRUB, sélectionnez l'entrée `Text Installer and command line`, puis l'option `3 Shell` dans l'écran d'installation en mode texte.
- SPARC - Programme d'installation automatisée : exécutez la commande suivante pour initialiser le système directement à partir d'un menu d'installation qui vous permet de quitter et d'accéder à un shell.

  ```
  ok boot net:dhcp
  ```
- x86 - Installation automatisée : l'initialisation à partir d'un serveur d'installation sur le réseau requiert une initialisation PXE. Sélectionnez l'entrée `Text Installer and command line` du menu GRUB. Sélectionnez ensuite l'option `3 Shell` à partir de l'écran d'installation en mode texte.

Par exemple, une fois que le système est initialisé, sélectionnez l'option `3 Shell`.

```
        1  Install Oracle Solaris
2  Install Additional Drivers
3  Shell
4  Terminal type (currently xterm)
5  Reboot

Please enter a number [1]: 3
To return to the main menu, exit the shell
#
```

2. Déterminez le problème impliquant une récupération par le biais d'une initialisation :

- Résolution d'un problème lié à une erreur de shell root en initialisant le système en mode monoutilisateur et correction de l'entrée shell dans le fichier `/etc/passwd`

  Sur un système x86, modifiez l'entrée d'initialisation sélectionnée et ajoutez l'option `-s`.

  Par exemple, sur un système SPARC, éteignez le système et initialisez en mode monoutilisateur. Une fois connecté en tant qu'utilisateur root, modifiez le fichier `/etc/passwd` et réparez l'entrée de shell root.

  ```
  # init 0
  ok boot -s
  Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a ...
  SunOS Release 5.11 Version 11.2 64-bit
  Copyright (c) 1983, 2013, Oracle and/or its affiliates. All rights reserved.
  Booting to milestone "milestone/single-user:default".
  Hostname: tardis.central
  Requesting System Maintenance Mode
  SINGLE USER MODE

  Enter user name for system maintenance (control-d to bypass): root
  Enter root password (control-d to bypass): xxxx
  ```

```
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode

Aug  3 15:46:21 su: 'su root' succeeded for root on /dev/console
Oracle Corporation      SunOS 5.11      11.2    July 2013
su: No shell /usr/bin/mybash.  Trying fallback shell /sbin/sh.
root@tardis.central:~# TERM =vt100; export TERM
root@tardis.central:~# vi /etc/passwd
root@tardis.central:~# <Press control-d>
logout
svc.startd: Returning to milestone all.
```

- Endommagé du chargeur de démarrage résoudre un problème.

  Tout d'abord, vous devez initialiser le système à partir d'un média ou du réseau à l'aide de l'une des méthodes d'initialisation répertoriées à l'étape 1. Ensuite, importez le pool root et corrigez une entrée GRUB, par exemple.

  ```
  x86# zpool import -f rpool
  ```

  Qu'ils soient réinstallés lors de l'aide du programme d'amorçage.

  ```
  x86# bootadm install-bootloader -f -P rpool
  ```

  où l'option `-f` force l'installation du programme d'amorçage et ignore les contrôles de version pour ne pas revenir à la version antérieure du programme d'amorçage sur le système. L'option `-p` permet d'indiquer le pool root.

  Vous pouvez utiliser la commande `bootadm list-menu` pour répertorier et modifier les entrées GRUB2. Pour plus d'informations, reportez-vous à `bootadm`(1M).

  Quitter et réinitialisez le système.

  ```
  x86# exit
  1  Install Oracle Solaris
  2  Install Additional Drivers
  3  Shell
  4  Terminal type (currently sun-color)
  5  Reboot

  Please enter a number [1]: 5
  ```

  Confirmez la réussite de l'initialisation du système.

- Résolvez un mot de passe root inconnu qui vous empêche de vous connecter au système.

  Tout d'abord, vous devez initialiser le système à partir d'un média ou du réseau à l'aide de l'une des méthodes d'initialisation répertoriées à l'étape 1. Ensuite, importez le pool root (`rpool`) et montez l'environnement d'initialisation afin de supprimer l'entrée de mot de passe root. Cette procédure est identique sur les plates-formes SPARC et x86.

```
# zpool import -f rpool
# beadm list
be_find_current_be: failed to find current BE name
be_find_current_be: failed to find current BE name
BE        Active Mountpoint Space  Policy Created
--        ------ ---------- -----  ------ -------
solaris   -      -          46.95M static 2012-07-20 10:25
solaris-2 R      -          3.81G  static 2012-07-19 13:44
# mkdir /a
# beadm mount solaris-2 /a
# TERM=vt100
# export TERM
# cd /a/etc
# vi shadow
<Carefully remove the unknown password>
# cd /
# beadm umount solaris-2
# halt
```

Accédez à l'étape suivante pour définir le mot de passe root.

3. **Définissez le mot de passe root en initialisant le système en mode monoutilisateur et en définissant le mot de passe.**

   Cette étape suppose que vous avez supprimé un mot de passe root inconnu à l'étape précédente.

   Sur un système x86, modifiez l'entrée d'initialisation sélectionnée et ajoutez l'option `-s`.

   Sur une plate-forme SPARC, initialisez le système en mode monoutilisateur, connectez-vous en tant qu'utilisateur root, puis définissez le mot de passe root. Par exemple :

```
ok boot -s
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a ...
SunOS Release 5.11 Version 11.2 64-bit
Copyright (c) 1983, 2013, Oracle and/or its affiliates. All rights reserved
Booting to milestone "milestone/single-user:default".

Enter user name for system maintenance (control-d to bypass): root
Enter root password (control-d to bypass): <Press return>
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode

Jul 20 14:09:59 su: 'su root' succeeded for root on /dev/console
Oracle Corporation      SunOS 5.11      11.2    July 2013
root@tardis.central:~# passwd -r files root
New Password: xxxxxx
Re-enter new Password: xxxxxx
passwd: password successfully changed for root
root@tardis.central:~# <Press control-d>
logout
```

```
svc.startd: Returning to milestone all.
```

♦ ♦ ♦ CHAPITRE 5

Gestion des systèmes de fichiers Oracle Solaris ZFS

Ce chapitre contient des informations détaillées sur la gestion des systèmes de fichiers ZFS Oracle Solaris. Il aborde notamment les concepts d'organisation hiérarchique des systèmes de fichiers, d'héritage des propriétés, de gestion automatique des points de montage et d'interaction sur les partages.

Ce chapitre contient les sections suivantes :

Gestion des systèmes de fichiers ZFS

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. Les systèmes de fichiers de leur légèreté et de leur rôle central dans l'administration du ZFS, il est fort probable que pour créer un grand nombre d'entre eux.

La gestion des systèmes de fichiers ZFS s'effectue à l'aide de la commande `zfs`. La commande `zfs` offre un ensemble de sous-commandes permettant d'effectuer des opérations spécifiques sur les systèmes de fichiers. Chacune de ces sous-commandes est décrite en détail dans ce

chapitre. Cette commande permet également de gérer les instantanés, les volumes et les clones. Toutefois, ces fonctionnalités sont uniquement traitées de manière succincte dans ce chapitre. Pour plus d'informations sur les instantanés et les clones, reportez-vous au Chapitre 6, Utilisation des instantanés et des clones ZFS Oracle Solaris. Pour plus d'informations sur les volumes ZFS, reportez-vous à la section “Volumes ZFS” à la page 269.

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

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 système de fichiers nom est spécifié en tant que le nom du chemin par rapport au nom du pool, comme suit :

pool-name/[filesystem-name/]filesystem-name

Le nom du pool et les noms des systèmes de fichiers existants mentionnés dans le chemin déterminent l'emplacement du nouveau système de fichiers dans la structure hiérarchique. Le dernier nom mentionné dans le chemin correspond au nom du système de fichiers à créer. Ce nom doit respecter les conventions d'attribution de nom définies à la section “Exigences d'attribution de noms de composants ZFS” à la page 18.

Le chiffrement d'un système de fichiers ZFS doit être activé au moment de sa création. Pour plus d'informations sur le chiffrement d'un système de fichiers ZFS, reportez-vous à la section “Chiffrement des systèmes de fichiers ZFS” à la page 191.

Dans l'exemple suivant, un système de fichiers nommé `ejeff` est créé dans le système de fichiers `tank/home`.

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

ZFS monte automatiquement le nouveau système de fichiers s'il est créé avec succès. Par défaut, les systèmes de fichiers sont montés sous */dataset*, à l'aide du chemin défini pour le nom du système dans la commande `create`. Dans cet exemple, le système de fichiers `jeff` créé est monté sous `/tank/home/jeff`. Pour plus d'informations sur les points de montage gérés automatiquement, reportez-vous à la section “Gestion des points de montage ZFS” à la page 168.

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 139.

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 169.

L'exemple suivant illustre la destruction du système de fichiers `tank/home/mark` :

```
# zfs destroy tank/home/mark
```

Attention - Aucune invite de confirmation ne s'affiche lors de l'exécution de la sous-commande `destroy`. Son utilisation requiert une attention particulière.

Si le système de fichiers à détruire est occupé et ne peut pas être démonté, la commande `zfs destroy` échoue. Pour détruire un système de fichiers actif, indiquez l'option `-f`. L'utilisation de cette option requiert une attention particulière. En effet, elle permet de démonter, d'annuler le partage et de détruire des systèmes de fichiers actifs, ce qui risque d'affecter le comportement de certaines applications.

```
# zfs destroy tank/home/matt
cannot unmount 'tank/home/matt': Device busy

# zfs destroy -f tank/home/matt
```

La commande `zfs destroy` échoue également si le système de fichiers possède des descendants. Pour détruire un système de fichiers et l'ensemble des descendants de ce

système de fichiers, indiquez l'option `-r`. Ce type d'opération de destruction récursive entraîne également la destruction des instantanés ; l'utilisation de cette option requiert donc une attention particulière.

```
# zfs destroy tank/ws
cannot destroy 'tank/ws': filesystem has children
use '-r' to destroy the following datasets:
tank/ws/jeff
tank/ws/bill
tank/ws/mark
# zfs destroy -r tank/ws
```

Si le système de fichiers à détruire possède des systèmes indirectement dépendants, même la commande de destruction récursive échoue. Pour forcer la destruction de *tous* les systèmes dépendants, y compris des systèmes de fichiers clonés situés en dehors de la structure hiérarchique cible, vous devez indiquer l'option `-R`. Utilisez cette option avec précaution.

```
# zfs destroy -r tank/home/eric
cannot destroy 'tank/home/eric': filesystem has dependent clones
use '-R' to destroy the following datasets:
tank//home/eric-clone
# zfs destroy -R tank/home/eric
```

Attention - Aucune invite de confirmation ne s'affiche lors de l'utilisation des options `-f`, `- r` ou `-R` avec la commande `zfs destroy`. L'utilisation de ces options requiert donc une attention particulière.

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

Modification du nom d'un système de fichiers ZFS

La modification du nom d'un système de fichiers ZFS s'effectue à l'aide de la commande `zfs rename`. Avec la sous-commande `rename`, vous pouvez effectuer les opérations suivantes :

- Modifier le nom d'un système de fichiers.
- Déplacer le système de fichiers au sein de la hiérarchie ZFS.
- Modifier le nom d'un système de fichiers et son emplacement au sein de la hiérarchie ZFS.

L'exemple suivant utilise la sous-commande `rename` pour renommer un système de fichiers `eric` en `eric_old` :

```
# zfs rename tank/home/eric tank/home/eric_old
```

L'exemple ci-dessous illustre la modification de l'emplacement d'un système de fichiers à l'aide de la sous-commande `zfs rename` :

```
# zfs rename tank/home/mark tank/ws/mark
```

Dans cet exemple, le système de fichiers `mark` est déplacé de `tank/home` vers `tank/ws`. Lorsque vous modifiez l'emplacement d'un système de fichiers à l'aide de la commande rename, le nouvel emplacement doit se trouver au sein du même pool et l'espace disque disponible doit être suffisant pour contenir le nouveau système de fichiers. Lorsque le nouvel emplacement ne dispose pas de suffisamment d'espace disque, l'opération `rename` échoue.

Pour plus d'informations sur les quotas, reportez-vous à la section “Définition des quotas et réservations ZFS” à la page 185.

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 206.

Présentation des propriétés ZFS

Les propriétés constituent le mécanisme principal de contrôle du comportement des systèmes de fichiers, des volumes, des instantanés et des clones. Sauf mention contraire, les propriétés définies dans cette section s'appliquent à tous les types de jeux de données.

Les propriétés se divisent en deux catégories : les propriétés natives et les propriétés définies par l'utilisateur. Les propriétés natives permettent de fournir des statistiques internes ou de contrôler le comportement du système de fichiers ZFS. Certaines de ces propriétés peuvent être définies tandis que d'autres sont en lecture seule. Les propriétés définies par l'utilisateur n'ont aucune incidence sur le comportement des systèmes de fichiers ZFS. En revanche, elles permettent d'annoter les jeux de données avec des informations adaptées à votre environnement. Pour plus d'informations sur les propriétés définies par l'utilisateur, reportez-vous à la section “Propriétés ZFS définies par l'utilisateur” à la page 158.

La plupart des propriétés pouvant être définies peuvent également être héritées. Les propriétés héritables sont des propriétés qui, une fois définies sur un système de fichiers parent, s'appliquent à l'ensemble de ses descendants.

Toutes les propriétés héritables sont associées à une source indiquant la façon dont la propriété a été obtenue. Les sources de propriétés peuvent avoir les valeurs suivantes :

`Local` — Indique que la propriété été définie de manière explicite sur le jeu de données à l'aide de la commande `zfs set`, selon la procédure décrite à la section “Paramétrage des propriétés ZFS” à la page 163.

`inherited from` *dataset-name* — Indique que la propriété a été héritée à partir de l'ascendant indiqué.

`Valeur par défaut` — Indique que la valeur de la propriété n'a été ni héritée, ni définie en local. Cette source est définie lorsque la propriété n'est pas définie en tant que source `local` sur aucun système ascendant.

Le tableau suivant répertorie les propriétés de système de fichiers ZFS natives en lecture seule et pouvant être définies. Les propriétés natives en lecture seule sont signalées comme tel. Les autres propriétés natives répertoriées dans le tableau peuvent être définies. Pour plus d'informations sur les propriétés définies par l'utilisateur, reportez-vous à la section “Propriétés ZFS définies par l'utilisateur” à la page 158.

TABLEAU 5-1 Description des propriétés ZFS natives

Nom de la propriété	Type	Valeur par défaut	Description
`aclinherit`	String	`Sécurisé`	Contrôle la manière dont les entrées ACL sont héritées quand les fichiers et répertoires sont créés. Les valeurs possibles sont `discard`, `noallow`, `secure` et `passthrough`. Pour une description de ces valeurs, reportez-vous à la section “Propriétés ACL” à la page 231.
`aclmode`	String	`groupmask`	Contrôle la manière dont une entrée ACL est modifiée pendant une opération `chmod`. Les valeurs possibles sont `discard`, `groupmask` et `passthrough`. Pour une description de ces valeurs, reportez-vous à la section “Propriétés ACL” à la page 231.
`atime`	Valeur booléenne	`on`	Vérifie si le temps d'accès des fichiers est mis à jour lors de la lecture de ces derniers. La désactivation de cette propriété évite de produire du trafic d'écriture lors de la lecture de fichiers et permet parfois d'améliorer considérablement les performances ; elle risque cependant de perturber les logiciels de messagerie et autres utilitaires du même type.
`available`	Nombre	S/O	Propriété en lecture seule indiquant la quantité d'espace disque disponible pour un système de fichiers et l'ensemble de ses enfants, sans tenir compte des autres activités du pool. L'espace disque étant partagé au sein d'un pool, l'espace disponible peut être limité par divers facteurs dont la taille du pool physique, les quotas, les réservations, ou les autres jeux de données présents au sein du pool. L'abréviation de la propriété est `avail`.

Nom de la propriété	Type	Valeur par défaut	Description
			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 20.
`canmount`	Valeur booléenne	`on`	Détermine si un système de fichiers donné peut être monté à l'aide de la commande `zfs mount`. Cette propriété peut être définie sur tous les systèmes de fichiers et ne peut pas être héritée. En revanche, lorsque cette propriété est définie sur `off`, un point de montage peut être hérité par des systèmes de fichiers descendants. Le système de fichiers à proprement parler n'est toutefois pas monté. Lorsque l'option `noauto` est définie, un système de fichiers peut uniquement être monté et démonté de manière explicite. Le système de fichiers n'est pas monté automatiquement lorsqu'il est créé ou importé, et il n'est pas monté par la commande `zfs mount-a` ni démonté par la commande `zfs unmount-a`. Pour plus d'informations, reportez-vous à la section “Propriété `canmount`” à la page 153.
`casesensitivity`	String	`mixed`	Cette propriété indique si l'algorithme de correspondance de nom de fichiers utilisé par le système de fichiers doit être `casesensitive` (respecter la casse), `caseinsensitive` (ne pas tenir compte de la casse) ou autoriser une combinaison des deux styles de correspondance (`mixed`). Traditionnellement, UNIX et systèmes de fichiers POSIX ont sensibles à la casse les noms de fichier. La valeur `mixed` de cette propriété indique que le système de fichiers peut prendre en charge les demandes pour le comportement de correspondance respectant la casse et pour le comportement de correspondance ne tenant pas compte de la casse. Actuellement, le comportement de correspondance ne tenant pas compte de la casse sur un système de fichiers prenant en charge un comportement mixte est limité au produit serveur SMB d'Oracle Solaris. Pour plus d'informations sur l'utilisation de la valeur `mixed`, reportez-vous à la section “Propriété `casesensitivity`” à la page 154. quel que soit le réglage de la propriété `casesensitivity`, le système de fichiers conserve la casse du nom indiqué pour créer un fichier. Cette propriété ne peut pas être modifiée après la création du système de fichiers.
`checksum`	String	`on`	Vérifie la somme de contrôle utilisée pour examiner l'intégrité des données. La valeur par défaut est définie sur `on`. Cette valeur permet de sélectionner automatiquement l'algorithme approprié, actuellement `fletcher4`. Les valeurs possibles sont `on, off, fletcher2, fletcher4, sha256` et `sha256+mac`.

Nom de la propriété	Type	Valeur par défaut	Description
			La valeur `off` entraîne la désactivation du contrôle d'intégrité des données utilisateur. La valeur `off` n'est pas recommandée.
`compression`	String	`off`	Activation ou désactivation de la compression pour un jeu de données. Les valeurs sont `on`, `off`, `lzjb`, `gzip` et `gzip-`*N*. Donner à cette propriété la valeur `lzjb`, `gzip` ou la valeur `gzip-` *N* a actuellement le même effet que la valeur `on`. L'activation de la compression sur un système de fichiers contenant des données existantes entraîne uniquement la compression des nouvelles données. Les données actuelles restent non compressées. L'abréviation de la propriété est `compress`.
`compressratio`	Nombre	S/O	Propriété en lecture seule indiquant le ratio de compression obtenu pour un jeu de données, exprimée sous forme de multiplicateur. La compression peut être activée moyennant la commande`zfs set compression=on` *dataset*. Cette valeur est calculée sur la base de la taille logique de l'ensemble des fichiers et de la quantité de données physiques indiquée. Elle induit un gain explicite basé sur l'utilisation de la propriété `compression`.
`copies`	Nombre	`1`	Définit le nombre de copies des données utilisateur par système de fichiers. Les valeurs disponibles sont `1`, `2` ou `3`. Ces copies viennent s'ajouter à toute redondance au niveau du pool. L'espace disque utilisé par plusieurs copies de données utilisateur est chargé dans le fichier et le jeu de données correspondants et pénalise les quotas et les réservations. En outre, la propriété `used` est mise à jour lorsque plusieurs copies sont activées. Considérez la définition de cette propriété à la création du système de fichiers car lorsque vous la modifiez sur un système de fichiers existant, les modifications ne s'appliquent qu'aux nouvelles données.
`creation`	String	S/O	Propriété en lecture seule indiquant la date et l'heure de la création d'un jeu de données.
`dedup`	String	`off`	Contrôle la possibilité de supprimer les données en double dans un système de fichiers ZFS. Les valeurs possibles sont `on`, `off`, `vérifier`, et `sha256[, vérifier]`. La somme de contrôle par défaut pour la suppression des doublons est `sha256`. Pour plus d'informations, reportez-vous à la section “Propriété `dedup`” à la page 155.
`devices`	Valeur booléenne	`on`	Contrôle si les fichiers du périphérique d'un système de fichiers peuvent être ouverts.
`encryption`	Valeur booléenne	`off`	Contrôle si un système de fichiers est chiffré. Un système de fichiers chiffré signifie que les données sont codées et qu'une clé est requise par le propriétaire du système de fichiers pour accéder aux données.

Nom de la propriété	Type	Valeur par défaut	Description
exec	Valeur booléenne	on	Contrôle si les programmes d'un système de fichiers sont autorisés pour exécution. Par ailleurs, lorsqu'elle est définie sur off, les appels mmap(2) avec PROT_EXEC ne sont pas autorisés.
keychangedate	String	Aucune	Identifie la date du dernier changement de clé d'encapsulation d'une opération zfs key -c pour le système de fichiers spécifié. Si aucune opération de changement de clé n'a été effectuée, la valeur de cette propriété en lecture seule est la même que la date de création du système de fichiers.
keysource	String	Aucune	Identifie le format et l'emplacement de la clé encapsulant les clés du système de fichiers. Les valeurs de propriété valides sont raw, hex, passphrase,prompt, ou *file*. La clé doit être présente lorsque le système de fichiers est créé, monté, ou chargé en utilisant la commande zfs key-l. Si le chiffrement est activé pour un nouveau système de fichiers, la keysource par défaut est passphrase,prompt.
keystatus	String	Aucune	Propriété en lecture seule identifiant le statut de la clé de chiffrement du système de fichiers. La disponibilité de la clé d'un système de fichiers est indiquée par disponible ou non disponible. Pour les systèmes de fichiers où le chiffrement n'est pas activé, none s'affiche.
logbias	String	Latence	Contrôle de quelle manière les requêtes synchronisée sont optimisées par ZFS pour ce système de fichiers. Si la propriété logbias est définie sur latency, ZFS utilise des périphériques de journalisation 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 journalisation 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.
mlslabel	String	Aucune.	Reportez-vous à la propriété multilevel pour obtenir une description du comportement de la propriété mlslabel sur des systèmes de fichiers multiniveau. La description de mlslabel suivante s'applique aux systèmes de fichiers multiniveau. Fournit une étiquette de sensibilité qui détermine si un système de fichiers peut être monté dans une zone Trusted Extensions. Si le système de fichiers étiqueté correspond à la zone étiquetée, le système de fichiers peut être monté et atteint depuis la zone étiquetée. La valeur par défaut est none. Cette propriété peut uniquement être modifiée lorsque Trusted Extensions est activé et que l'utilisateur dispose du privilège approprié.
mounted	Valeur booléenne	S/O	Propriété en lecture indiquant si un système de fichiers clone ou instantané est actuellement monté. Cette

Nom de la propriété	Type	Valeur par défaut	Description
			propriété ne s'applique pas aux volumes. Les valeurs possibles sont `yes` ou `no`.
`mountpoint`	String	S/O	Contrôle le point de montage utilisé pour ce système de fichiers. Lorsque la propriété `mountpoint` (point de montage) 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 168.
`multilevel`	Valeur booléenne	`off`	Cette propriété peut uniquement être utilisée avec Trusted Extensions activé. La valeur par défaut est off. Les objets dans un système de fichiers multiniveau sont marqués individuellement avec un attribut d'étiquette de sensibilité explicite qui est automatiquement généré. Il est possible de changer l'étiquette des objets en changeant l'attribut d'étiquette à l'aide des interfaces `setlabel` ou `setflabel`. Les systèmes de fichiers root, Oracle Solaris Zone ou contenant du code Solaris ne doivent pas être multiniveau. Des différences existent dans la propriété `mlslabel` sur un système de fichiers multiniveau. La valeur `mlslabel` définit l'étiquette la plus élevée possible pour les objets dans le système de fichiers. Une tentative de création d'un fichier à (ou de changement d'étiquette d'un fichier vers) une étiquette plus élevée que la valeur `mlslabel` n'est pas autorisée. Une stratégie de montage basée sur la valeur `mlslabel` ne s'applique pas à un système de fichiers multiniveau. Pour un système de fichiers multiniveau, la propriété `mlslabel` peut être définie explicitement à la création du système de fichiers. Sinon, une propriété `mlslabel` par défaut de `ADMIN_HIGH` est automatiquement créée. Après la création d'un système de fichiers multiniveau, la propriété `mlslabel` peut être modifiée, mais elle ne peut pas être définie sur un niveau inférieur, sur `none` ni être supprimée.
`primarycache`	String	`Toutes`	Contrôle le contenu de la cache principale (ARC). Les valeurs possibles sont `all`, `none` et `metadata`. Si elles sont définies sur `all`, les données d'utilisateur

Nom de la propriété	Type	Valeur par défaut	Description
			et les métadonnées sont mises en cache. Si elle est définie sur `none`, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si elles sont définies sur `metadata`, seules les métadonnées sont mises en cache. Lorsque ces propriétés sont définies sur des systèmes de fichiers existants, seule la nouvelle E/S est mise en cache en fonction de la valeur de ces propriétés. Certains environnements de bases de données pourraient bénéficier de la non-mise en cache des données d'utilisateur. Vous devrez déterminer si la définition des propriétés du cache est appropriée pour votre environnement.
`nbmand`	Valeur booléenne	`off`	Contrôle si le système de fichiers doit être monté avec des verrous `nbmand` (obligatoires non bloquants). Cette propriété est réservée aux clients SMB. Les modifications apportées à cette propriété s'appliquent uniquement lorsque le système de fichiers est démonté puis remonté.
`normalization`	String	Aucune.	Cette propriété indique si un système de fichiers doit effectuer une normalisation Unicode des noms de fichiers dès lors que deux noms de fichier sont comparés, et précise l'algorithme de normalisation à utiliser. Les noms de fichier sont toujours stockés sans être modifiés, et sont normalisés dans le cadre de n'importe quel processus de comparaison. Si la valeur de cette propriété est définie sur une valeur légale autre que `none` et que la propriété `utf8only` n'est pas renseignée, la propriété `utf8only` est automatiquement définie sur `on`. La valeur par défaut de la propriété `normalisation` est `aucune`. Cette propriété ne peut pas être modifiée après la création du système de fichiers.
`origin`	String	S/O	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 ont une origine `none` (aucune).
`quota`	Valeur numérique (ou `none` (aucune))	`none`	Limite la quantité d'espace disque qu'un système de fichiers et ses descendants peuvent consommer. Cette propriété permet d'appliquer une limite fixe à la quantité d'espace disque utilisée, y compris l'espace utilisé par les descendants, qu'il s'agisse de systèmes de fichiers ou d'instantanés. La définition d'un quota sur un descendant de système de fichiers déjà associé à un quota n'entraîne pas le remplacement du quota du système ascendant. Cette opération entraîne au contraire l'application d'une limite supplémentaire. Les quotas ne peuvent pas être définis pour les volumes, car la propriété `volsize` sert de quota implicite.

Nom de la propriété	Type	Valeur par défaut	Description
			Pour plus d'informations sur la définition de quotas, reportez-vous à la section “Définitions de quotas sur les systèmes de fichiers ZFS” à la page 186.
`rekeydate`	String	S/O	Propriété en lecture seule indiquant la date de la dernière modification de la clé de chiffrement des données résultant d'une opération `zfs key -K` ou `zfs clone -K` sur ce système de fichiers. Si aucune opération `rekey` n'a été effectuée, la valeur de cette propriété est identique à la date de `creation`.
`readonly`	Valeur booléenne	`off`	Contrôle si un jeu de données peut être modifié. Lorsqu'elle est définie sur `on`, aucune modification ne peut être apportée. L'abréviation de la propriété est `rdonly`.
`recordsize`	Nombre	`128K`	Spécifie un taille de bloc pour les fichiers d'un système de fichiers. L'abréviation de la propriété est `recsize`. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété `recordsize` ” à la page 157.
`referenced`	Nombre	S/O	Propriété en lecture seule identifiant la quantité de données à laquelle un jeu de données a accès, lesquelles peuvent ou non être partagées avec d'autres jeux de données du pool. Lorsqu'un instantané ou un clone est créé, il indique dans un premier temps la même quantité d'espace disque que le système de fichiers ou l'instantané à partir duquel il a été créé. En effet, son contenu est identique. L'abréviation de la propriété est `refer`.
`refquota`	Valeur numérique (ou none)	`none`	Définit la quantité d'espace disque qu'un jeu de données peut occuper. Cette propriété définit une quantité d'espace maximale. Cette limite fixe n'inclut pas l'espace disque utilisé par les descendants, tels que les instantanés et les clones.
`refreservation`	Valeur numérique (ou none)	`none`	Définit la quantité d'espace disque minimale garantie pour un jeu de données, à l'exclusion des descendants, notamment les instantanés et les clones. Lorsque la quantité d'espace disque utilisée est inférieure à cette valeur, le système considère que le jeu de donnée utilise la quantité d'espace spécifiée par `refreservation` La réservation `refreservation` est prise en compte dans l'espace disque utilisé des jeux de donnés parent et vient en déduction de leurs quotas et réservations. Lorsque la propriété `refreservation` est définie, un instantané n'est autorisé que si suffisamment d'espace est disponible dans le pool au-delà de cette réservation afin de pouvoir contenir le nombre actuel d'octets *référencés* dans le jeu de données.

Nom de la propriété	Type	Valeur par défaut	Description
			L'abréviation de la propriété est `refserv`.
`reservation`	Valeur numérique (ou none)	`none`	Définit les ZFS la quantité d'espace disque minimale garantie à un système de fichiers et ses descendants. Lorsque la quantité d'espace disque utilisée est inférieure à cette valeur, le système considère que le système de fichiers utilise la quantité d'espace réservée. Les réservations sont prises en compte dans l'espace disque utilisé du système de fichiers parent, et viennent en déduction des quotas et réservations de celui-ci. L'abréviation de la propriété est `reserv`. Pour plus d'informations, reportez-vous à la section “Définition de réservations sur les systèmes de fichiers ZFS” à la page 189.
`rstchown`	Valeur booléenne	le	Indique si le propriétaire du système de fichiers peut autoriser la modification du ou des propriétaires de fichiers. Par défaut, les opérations `chown` sont restreintes. Lorsque `rstchown` est défini sur off, l'utilisateur dispose du privilège `PRIV_FILE_CHOWN_SELF` pour les opérations `chown`.
`secondarycache`	String	`Toutes`	Contrôle le contenu de la cache secondaire (L2ARC). Les valeurs possibles sont `all`, `none` et `metadata`. Si elles sont définies sur `all`, les données d'utilisateur et les métadonnées sont mises en cache. Si elle est définie sur `none`, ni les données d'utilisateur ni les métadonnées ne sont mises en cache. Si elles sont définies sur `metadata`, seules les métadonnées sont mises en cache.
`setuid`	Valeur booléenne	`on`	Contrôle si le bit `setuid` est respecté dans un système de fichiers.
`shadow`	String	`None`	Identifie un système de fichiers ZFS en tant que *shadow* du système de fichiers décrit par l'*URI*. Les données sont migrées vers un système de fichiers shadow pour lequel cette propriété est activée à partir du système de fichiers identifié par l'URI. Le système de fichiers à migrer doit être en lecture seule pour une migration complète.
`share.nfs`	String	`off`	Détermine si un partage pour un système de fichiers ZFS est créé et publié, et quelles options sont utilisées. Vous pouvez également publier et annuler la publication d'un partage NFS à l'aide des commandes `zfs share` et `zfs unshare`. L'utilisation de la commande `zfs share` pour publier un partage NFS nécessite qu'une propriété de partage NFS soit également définie. Pour plus d'informations sur la définition des propriétés de partage NFS, reportez-vous à la section “Activation et annulation du partage des systèmes de fichiers ZFS” à la page 173. Pour plus d'informations sur le partage des systèmes de fichiers ZFS, reportez-vous à la section “Activation et annulation du partage des systèmes de fichiers ZFS” à la page 173.

Nom de la propriété	Type	Valeur par défaut	Description
`share.smb`	String	`off`	Contrôle si un partage SMB d'un système de fichiers ZFS est créé et publié, et quelles options sont utilisées. Vous pouvez également publier et annuler la publication d'un partage SMB à l'aide des commandes `zfs share` et `zfs unshare`. L'utilisation de la commande `zfs share` pour publier un partage SMB nécessite qu'une propriété de partage SMB soit également définie. Pour plus d'informations sur la définition des propriétés de partage SMB, reportez-vous à la section "Activation et annulation du partage des systèmes de fichiers ZFS" à la page 173.
`snapdir`	String	`hidden`	Contrôle si le répertoire `.zfs` est affiché ou masqué au niveau root du système de fichiers. Pour plus d'informations sur l'utilisation des instantanés, reportez-vous à la section "Présentation des instantanés ZFS" à la page 203.
`sync`	String	`Standard`	Détermine le comportement synchronisé des transactions d'un système de fichiers. Les valeurs possibles sont les suivantes : ■ `standard`, la valeur par défaut, avec laquelle les transactions des systèmes de fichiers synchrones, telles que `fsync`, `O_DSYNC`, `O_SYNC`, etc. sont consignées dans le journal d'intention. ■ `always`, qui garantit que *chaque* transaction de système de fichiers est consignée et validée sur le stockage stable par un appel système retourné. Cette valeur réduit sensiblement les performances. ■ `disabled`, signifie que les demandes synchrones sont désactivées. Les transactions du système de fichiers sont uniquement validées sur le stockage stable lors de la validation de groupe de transactions suivante, qui peut intervenir après un grand nombre de secondes. Cette valeur permet les meilleures performances, et tout risque d'endommagement du pool est exclu. **Attention -** Cette valeur `disabled` est très dangereuse car le système de fichiers ZFS ignore les demandes de transaction synchrones des applications, telles que les bases de données ou les opérations NFS. Le réglage de cette valeur sur le système de fichiers root ou `/var` actuellement actif peut entraîner un comportement inattendu, une perte des données de l'application ou un accroissement de la vulnérabilité aux attaques en boucle. N'utilisez cette valeur que si vous parfaitement averti de tous les risques associés.
`type`	String	S/O	Propriété en lecture seule identifiant le type de jeu de données comme `filesystem` (système de fichiers ou clone), `volume`, ou `snapshot` (instantané).

Nom de la propriété	Type	Valeur par défaut	Description
`used`	Nombre	S/O	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par le jeu de données et l'ensemble de ses descendants. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété `used`” à la page 151.
`usedbychildren`	Nombre	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par les enfants de ce jeu de données, qui serait libérée si tous ses enfants étaient détruits. L'abréviation de la propriété est `usedchild`.
`usedbydataset`	Nombre	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par le jeu de données lui-même, qui serait libérée si ce dernier était détruit, après la destruction préalable de tous les instantanés et la suppression de toutes les réservations `refreservation`. L'abréviation de la propriété est `usedds`.
`usedbyrefreservation`	Nombre	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par un jeu `refreservation` sur un jeu de données, qui serait libérée si le jeu `refreservation` était supprimé. L'abréviation de la propriété est `usedrefreserv`.
`usedbysnapshots`	Nombre	`off`	Propriété en lecture seule indiquant la quantité d'espace disque utilisée par les instantanés de ce jeu de données. En particulier, elle correspond à la quantité d'espace disque qui serait libérée si l'ensemble des instantanés de ce jeu de données était supprimé. Notez que cette valeur ne correspond pas simplement à la somme des propriétés `used` des instantanés, car l'espace peut être partagé par plusieurs instantanés. L'abréviation de la propriété est `usedsnap`.
`version`	Nombre	S/O	Identifie la version du disque d'un système de fichiers, indépendamment de la version du pool. Cette propriété peut uniquement être définie avec une version supérieure prise en charge par la version du logiciel. Pour plus d'informations, reportez-vous à la commande. `zfs upgrade`
`utf8only`	Valeur booléenne	`Off`	Cette propriété indique si un système de fichiers doit rejeter des noms de fichiers contenant des caractères non inclus dans l'ensemble de caractères au format UTF-8. Si cette propriété est définie de façon explicite sur `off`, la propriété `normalization` doit soit être définie de façon non explicite soit être définie sur `none` . La valeur par défaut pour la propriété `utf8only` est `off`. Cette propriété ne peut pas être modifiée après la création du système de fichiers.
`volsize`	Nombre	S/O	Spécifie, pour les volumes, la taille logique du volume. Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété `volsize`” à la page 158.

Nom de la propriété	Type	Valeur par défaut	Description
`volblocksize`	Nombre	`8 KO`	Spécifie, pour les volumes, la taille du 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. N'importe quelle puissance comprise entre 2 et 512 octets et 128 KO est valide. L'abréviation de la propriété est `volblock`.
`vscan`	Valeur booléenne	`Off`	Contrôle si des recherches de virus doivent être effectuées sur les fichiers standard lorsqu'un fichier est ouvert et fermé. En plus de cette propriété, un service d'analyse antivirus doit également être activé afin de permettre le lancement des analyses sur des logiciels antivirus tiers, si vous en possédez. La valeur par défaut est `off`.
`zoned`	Valeur booléenne	S/O	Indique si un système de fichiers 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 demandes. Lors de la première installation d'une zone, cette propriété est définie pour tous les systèmes de fichiers ajoutés. Pour plus d'informations sur l'utilisation du système ZFS avec des zones installées, reportez-vous à la section “Utilisation de ZFS dans un système Solaris avec zones installées” à la page 272.
`xattr`	Valeur booléenne	`on`	Indique si les attributs étendus sont activés (`on`) ou désactivés (`off`) pour ce système de fichiers.

Propriétés ZFS natives en lecture seule

Les propriétés natives en lecture seule peuvent être récupérées, mais ne peuvent pas être définies. Elles ne peuvent pas non plus être héritées. Certaines propriétés natives sont spécifiques à un type de jeu de données. Dans ce cas, le type de jeu de données est mentionné dans la description figurant dans le Tableau 5-1, “Description des propriétés ZFS natives”.

Les propriétés natives en lecture seule sont répertoriées dans cette section et décrites dans le Tableau 5-1, “Description des propriétés ZFS natives”.

- `available`
- `compressratio`
- `création`
- `keystatus`
- `mounted`

- `origin`
- `referenced`
- `rekeydate`
- `type`
- `used`

  Pour plus d'informations sur cette propriété, reportez-vous à la section “Propriété used” à la page 151.
- `usedbychildren`
- `usedbydataset`
- `usedbyrefreservation`
- `usedbysnapshots`

Pour plus d'informations sur la détermination de l'espace disque, notamment sur les propriétés `used`, `referenced` et `available`, reportez-vous à la section “Comptabilisation de l'espace disque ZFS” à la page 20.

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 21.

La quantité d'espace disque utilisé, disponible et référencé ne comprend pas les modifications en attente. Ces modifications sont généralement prises en compte au bout de quelques secondes. La modification d'un disque utilisant la fonction `fsync(3c)` ou `O_SYNC` ne garantit pas la mise à jour immédiate des informations concernant l'utilisation de l'espace disque.

Les informations de propriété `usedbychildren`, `usedbydataset` , `usedbyrefreservation` et `usedbysnapshots` peuvent être affichées à l'aide de la commande `zfs list-o space`. Ces

propriétés divisent la propriété `used` en espace disque utilisé par les descendants. Pour plus d'informations, reportez-vous au Tableau 5-1, “Description des propriétés ZFS natives”.

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 “Paramétrage des propriétés ZFS” à la page 163 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 136. A l'exception des quotas et des réservations, les propriétés natives définies sont héritées. Pour plus d'informations sur les quotas, reportez-vous à la section “Définition des quotas et réservations ZFS” à la page 185.

Certaines propriétés natives définies sont spécifiques à un type de jeu de données. Dans ce cas, le type de jeu de données est mentionné dans la description figurant dans le Tableau 5-1, “Description des propriétés ZFS natives”. Sauf indication contraire, les propriétés s'appliquent à tous les types de jeu de données : aux systèmes de fichiers, aux volumes, aux clones et aux instantanés.

Les propriétés pouvant être définies sont répertoriées dans cette section et décrites dans le Tableau 5-1, “Description des propriétés ZFS natives”.

- `aclinherit`

  Pour obtenir une description détaillée, reportez-vous à la section “Propriétés ACL” à la page 231.
- `aclmode`

  Pour obtenir une description détaillée, reportez-vous à la section “Propriétés ACL” à la page 231.
- `atime`
- `canmount`
- `casesensitivity`
- `checksum`
- `compression`
- `copies`
- `périphériques`
- `dedup`
- `encryption`
- `exec`
- `keysource`
- `logbias`

- `mlslabel`
- `mountpoint`
- `nbmand`
- `normalisation`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`

  Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété `recordsize`” à la page 157.
- `refquota`
- `refreservation`
- `reservation`
- `rstchown`
- `secondarycache`
- `share.smb`
- `share.nfs`
- `setuid`
- `snapdir`
- `version`
- `vscan`
- `utf8only`
- `volsize`

  Pour obtenir une description détaillée de cette propriété, reportez-vous à la section “Propriété `volsize`” à la page 158.
- `volblocksize`
- `zoned`
- `xattr`

Propriété `canmount`

Si la propriété `canmount` est désactivée (valeur `off`), le système de fichiers ne peut pas être monté à l'aide de la commande `zfs mount`, ni de la commande `zfs mount-a`. Définir cette propriété sur `off` équivaut à définir la propriété `mountpoint` sur `none`, à la différence près que le système de fichiers conserve une propriété `mountpoint` normale pouvant être héritée. Vous pouvez par exemple définir cette propriété sur la valeur `off` et définir des propriétés héritées pour les systèmes de fichiers descendants. Toutefois, le système de fichiers parent à proprement

parler n'est jamais monté, ni accessible par les utilisateurs. Dans ce cas, le système de fichiers parent sert de *container* 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
```

Lorsque la propriété `canmount` est définie sur `noauto`, le système de fichiers peut uniquement être monté de façon explicite et pas automatique.

Propriété `casesensitivity`

Cette propriété indique si l'algorithme de correspondance de nom de fichiers utilisé par le système de fichiers doit être `casesensitive` (respecter la casse), `caseinsensitive` (ne pas tenir compte de la casse) ou autoriser une combinaison des deux styles de correspondance (`mixed`).

Lorsqu'une demande de correspondance ne tenant pas compte de la casse est effectuée sur un système de fichiers défini sur *mixed*, le comportement est généralement identique à ce qu'il serait sur un système de fichiers ne tenant pas compte de la casse. Toutefois, un système de fichiers doté d'une sensibilité à la casse "mixte" peut contenir des répertoires portant des noms uniques en cas de respect de la casse, mais qui ne sont pas uniques lorsque la casse n'est pas prise en compte.

Par exemple, un répertoire peut contenir des fichiers nommés `foo`, `Foo` et `FOO`. Si une demande de correspondance ne tenant pas compte de la casse est effectuée pour l'une des formes possibles de `foo` (par exemple `foo`, `FOO`, `FoO`, `fOo`, etc.), l'algorithme de correspondance sélectionne en tant que correspondance l'un des trois fichiers existants. Il est impossible de prévoir avec certitude quel fichier sera choisi comme correspondance par l'algorithme ; en revanche, il est certain que le même fichier sera choisi comme correspondance pour toutes les formes de `foo`. Le fichier choisi comme correspondance ne tenant pas compte de la casse pour `foo`, `FOO` , `foO`, `Foo`, et ainsi de suite, est toujours le même, tant que le répertoire n'est pas modifié.

Les propriétés `utf8only`, `normalization` et `casesensitivity` fournissent également de nouvelles autorisations qui peuvent être attribuées à des utilisateurs non privilégiés par le biais de l'administration déléguée de ZFS. Pour plus d'informations, reportez-vous à la section “Délégation d'autorisations ZFS” à la page 256.

Propriété `copies`

A des fins de fiabilité, les métadonnées d'un système de fichiers ZFS sont automatiquement stockées plusieurs fois sur différents disques, lorsque cela est 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`. Par exemple :

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

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

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

- Cela améliore la rétention des données en autorisant leur récupération à partir d'erreurs de lecture de blocs irrécupérables, comme par exemple des défaillances de média (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.

Propriété `dedup`

La propriété `dedup` détermine si les données en double sont supprimées d'un système de fichiers. Si un système de fichiers a la `dédupliquer` propriété activée, blocs de données dupliquées sont supprimés simultanément. Par conséquent, seules les données uniques sont stockées et les composants communs sont partagés entre les fichiers.

N'activez pas la propriété `dedup` sur des systèmes de fichiers résidant sur des systèmes de production avant d'avoir passé en revue les points suivants :

1. Déterminez si vous pouvez économiser de l'espace grâce à la suppression des doublons. Vous pouvez exécuter la commande `zdb -S` pour simuler les sommes des gains sur l'activation de dedup potentiels d'espace sur votre pool. Cette commande doit être exécutée sur un lieu silencieux pool. Si la suppression des doublons ne s'appliquent pas à vos données, inutile d'activer cette propriété. Par exemple :

```
# zdb -S tank
Simulated DDT histogram:
bucket              allocated                       referenced
______   ______________________________   ______________________________
refcnt   blocks   LSIZE   PSIZE   DSIZE   blocks   LSIZE   PSIZE   DSIZE
------   ------   -----   -----   -----   ------   -----   -----   -----
1    2.27M    239G    188G    194G    2.27M    239G    188G    194G
2     327K   34.3G   27.8G   28.1G     698K   73.3G   59.2G   59.9G
4    30.1K   2.91G   2.10G   2.11G     152K   14.9G   10.6G   10.6G
8    7.73K    691M    529M    529M    74.5K   6.25G   4.79G   4.80G
16       673   43.7M   25.8M   25.9M    13.1K    822M    492M    494M
32       197   12.3M   7.02M   7.03M    7.66K    480M    269M    270M
64        47   1.27M    626K    626K    3.86K    103M   51.2M   51.2M
128       22    908K    250K    251K    3.71K    150M   40.3M   40.3M
256        7    302K     48K   53.7K    2.27K   88.6M   17.3M   19.5M
512        4    131K   7.50K   7.75K    2.74K    102M   5.62M   5.79M
2K         1      2K      2K      2K    3.23K   6.47M   6.47M   6.47M
8K         1    128K      5K      5K    13.9K   1.74G   69.5M   69.5M
Total    2.63M    277G    218G    225G    3.22M    337G    263G    270G

dedup = 1.20, compress = 1.28, copies = 1.03, dedup * compress / copies = 1.50
```

   Si le ratio estimé est supérieur à 2, la suppression des doublons est susceptible de vous faire gagner de la place.

   Dans l'exemple ci-dessus, le ratio de suppression des doublons est inférieur à 2, si bien que l'activation de dedup n'est pas recommandée.

2. Assurez-vous que votre système dispose de suffisamment de mémoire pour prendre en charge dedup.
   - Chaque entrée de table dedup interne a une taille d'environ 320 octets.
   - Multipliez le nombre de blocs alloués par 320. Par exemple :

     ```
     in-core DDT size = 2.63M x 320 = 841.60M
     ```

3. Les performances de la propriété dedup sont optimisées lorsque le tableau de suppression des doublons tient en mémoire. Si ce tableau doit être écrit sur le disque, les performances diminuent. Par exemple, la suppression d'un système de fichiers volumineux lorsque l'option de dépuplication est activée entrave considérablement les performances du système si les conditions relatives à la mémoire évoquées plus haut ne sont pas satisfaites.

Quand dedup est activé, l'algorithme de somme de contrôle dedup écrase la propriété checksum. Définir la valeur de propriété sur verify équivaut à spécifier sha256,verify. Si la propriété est définie sur verify et que deux blocs ont la même signature, ZFS effectue une vérification octet par octet avec le bloc existant afin de garantir que les contenus sont identiques.

Cette propriété peut être activée pour chaque système de fichiers. Par exemple :

```
# zfs set dedup=on tank/home
```

Vous pouvez utiliser la commande zfs get pour déterminer si la propriété dedup est définie.

Bien que la suppression des doublons soit définie en tant que propriété du système de fichiers, elle s'étend à l'échelle du pool. Par exemple, vous pouvez identifier le ratio de suppression des doublons. Par exemple :

```
# zpool list tank
NAME    SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
rpool   136G  55.2G  80.8G    40%  2.30x  ONLINE  -
```

La colonne DEDUP indique le nombre de suppressions de doublons effectuées. Si la propriété dedup n'est activée sur aucun système de fichiers ou si la propriété dedup vient d'être activée sur le système de fichiers, le ratio DEDUP est 1.00x.

Vous pouvez utiliser la commande zpool get pour déterminer la valeur de la propriété dedupratio. Par exemple :

```
# zpool get dedupratio export
NAME   PROPERTY    VALUE  SOURCE
rpool  dedupratio  3.00x  -
```

Cette propriété du pool illustre le nombre de suppressions de doublons de données effectuées dans ce pool.

Propriété encryption

Vous pouvez utiliser la propriété encryption pour chiffrer les systèmes de fichiers ZFS. Pour plus d'informations, reportez-vous à la section “Chiffrement des systèmes de fichiers ZFS” à la page 191.

Propriété recordsize

La propriété recordsize spécifie une taille de bloc suggérée pour les fichiers du système de fichiers.

Cette propriété s'utilise uniquement pour les charges de travail de base de données accédant à des fichiers résidant dans des enregistrements à taille fixe. Le système ZFS ajuste automatiquement les tailles en fonction d'algorithmes internes optimisés pour les schémas

d'accès classiques. Pour les bases de données générant des fichiers volumineux mais accédant uniquement à certains fragments de manière aléatoire, ces algorithmes peuvent se révéler inadaptés. La définition d'une valeur `recordsize` supérieure ou égale à la taille d'enregistrement de la base de données peut améliorer les performances du système de manière significative. Il est vivement déconseillé d'utiliser cette propriété pour les systèmes de fichiers à usage générique. En outre, elle peut affecter les performances du système. La taille spécifiée doit être une puissance de 2 supérieure ou égale à 512 octets et inférieure ou égale à 1 MO. 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 `share.smb`

Cette propriété permet de partager des systèmes de fichiers ZFS avec le service Oracle Solaris SMB et d'identifier les options à utiliser.

Quand la propriété est activée, tous les partages qui héritent de la propriété sont repartagés avec leurs options actuelles. Quand la propriété est désactivée, les partages qui héritent de la propriété cessent d'être partagés.Pour obtenir des exemples d'utilisation de la propriété `share.smb`, reportez-vous à la section “Activation et annulation du partage des systèmes de fichiers ZFS” à la page 173.

Propriété `volsize`

La propriété `volsize` spécifie la taille logique du volume. Par défaut, la création d'un volume définit une réservation de taille identique. Toute modification apportée à la valeur de la propriété `volsize` se répercute dans des proportions identiques au niveau de la réservation. Ce fonctionnement permet d'éviter les comportements inattendus lors de l'utilisation des volumes. L'utilisation de volumes contenant moins d'espace disponible que la valeur indiquée risque, suivant le cas, d'entraîner des comportements non valides et des altérations de données. Ces symptômes peuvent également survenir lors de la modification et notamment de la réduction de la taille du volume en cours d'utilisation. Faites preuve de prudence lorsque vous ajustez la taille d'un volume.

Pour plus d'informations sur l'utilisation des volumes, reportez-vous à la section “Volumes ZFS” à la page 269.

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.

Il doit respecter les conventions suivantes :

- Elles doivent contenir le caractère ":" (deux points) afin de les distinguer des propriétés natives.
- Elles doivent contenir des lettres en minuscule, des chiffres ou les signes de ponctuation suivants : ':', '+','.', '_'.
- La longueur maximale du nom d'une propriété définie par l'utilisateur est de 256 caractères.

La syntaxe attendue des noms de propriétés consiste à regrouper les deux composants suivants (cet espace de noms n'est toutefois pas appliqué par les systèmes ZFS) :

module:property

Si vous utilisez des propriétés définies par l'utilisateur dans un contexte de programmation, spécifiez un nom de domaine DNS inversé pour le composant *module* des noms de propriétés, afin de réduire la probabilité que deux packages développés séparément n'utilisent un nom de propriété identique à des fins différentes. Les noms de propriété commençant par `com.oracle.` sont réservés à l'usage d'Oracle Corporation.

Les valeurs des propriétés définies par l'utilisateur doivent respecter les conventions suivantes :

- Elles doivent être constituées de chaînes arbitraires systématiquement héritées et elle ne doivent jamais être validées.
- La longueur maximale de la valeur d'une propriété définie par l'utilisateur est de 1024 caractères.

Par exemple :

```
# zfs set dept:users=finance userpool/user1
# zfs set dept:users=general userpool/user2
# zfs set dept:users=itops userpool/user3
```

Toutes les commandes fonctionnant avec des propriétés (par exemple, les commandes `zfs list`, `zfs get`, `zfs set`, etc.) permettent d'utiliser des propriétés natives et des propriétés définies par l'utilisateur.

Par exemple :

```
zfs get -r dept:users userpool
NAME            PROPERTY    VALUE           SOURCE
userpool        dept:users  all             local
userpool/user1  dept:users  finance         local
userpool/user2  dept:users  general         local
userpool/user3  dept:users  itops           local
```

Pour supprimer une propriété définie par l'utilisateur, utilisez la commande `zfs inherit`. Par exemple :

```
# zfs inherit -r dept:users userpool
```

Si cette propriété n'est définie dans aucun jeu de données parent, elle est définitivement supprimée.

Envoi de requêtes sur les informations des systèmes de fichiers ZFS

La commande `zfs list` contient un mécanisme extensible permettant d'afficher et d'envoyer des requêtes sur les informations des systèmes de fichiers. Cette section décrit les requêtes de base ainsi que les requêtes plus complexes.

Affichage des informations de base des systèmes ZFS

La commande `zfs list` spécifiée sans option permet de répertorier les informations de base sur les jeux de données. Cette commande affiche le nom de tous les jeux de données définis sur le système ainsi que les valeurs `used`, `available`, `referenced` et `mountpoint` correspondantes. Pour plus d'informations sur les propriétés, reportez-vous à la section “Présentation des propriétés ZFS” à la page 139.

Par exemple :

```
# zfs list
users                  2.00G  64.9G    32K  /users
users/home             2.00G  64.9G    35K  /users/home
users/home/cindy        548K  64.9G   548K  /users/home/cindy
users/home/mark        1.00G  64.9G  1.00G  /users/home/mark
users/home/neil        1.00G  64.9G  1.00G  /users/home/neil
```

Cette commande permet d'afficher des jeux de données spécifiques. Pour cela, spécifiez le nom du ou des jeux de données à afficher sur la ligne de commande. Vous pouvez également spécifier l'option `-r` pour afficher de manière récursive tous les descendants des jeux de données. Par exemple :

```
# zfs list -t all -r users/home/mark
NAME                       USED  AVAIL  REFER  MOUNTPOINT
users/home/mark           1.00G  64.9G  1.00G  /users/home/mark
users/home/mark@yesterday     0      -  1.00G  -
users/home/mark@today         0      -  1.00G  -
```

Vous pouvez utiliser la commande `zfs list` avec le point de montage d'un système de fichiers. Par exemple :

```
# zfs list /user/home/mark
NAME             USED  AVAIL  REFER  MOUNTPOINT
```

```
users/home/mark  1.00G  64.9G  1.00G  /users/home/mark
```

L'exemple ci-dessous montre comment afficher des informations de base sur `tank/home/gina` et tous ses systèmes de fichiers descendants :

```
# zfs list -r users/home/gina
NAME                          USED  AVAIL  REFER  MOUNTPOINT
users/home/gina               2.00G  62.9G    32K  /users/home/gina
users/home/gina/projects      2.00G  62.9G    33K  /users/home/gina/projects
users/home/gina/projects/fs1  1.00G  62.9G  1.00G  /users/home/gina/projects/fs1
users/home/gina/projects/fs2  1.00G  62.9G  1.00G  /users/home/gina/projects/fs2
```

Pour plus d'informations sur la commande `zfs list`, reportez-vous à la page de manuel `zfs`(1M).

Création de requêtes ZFS complexes

Les options `o`, `-t` et `-H` permettent de personnaliser la sortie de la commande `-zfs list`.

Vous pouvez également personnaliser la sortie des valeurs de propriété en spécifiant l'option `-o` ainsi que la liste des propriétés souhaitées séparées par une virgule. Toute propriété de jeu de données peut être utilisée en tant qu'argument valide. Pour consulter la liste de toutes les propriétés de jeu de données prises en charge, reportez-vous à la section “Présentation des propriétés ZFS” à la page 139. Outre les propriétés répertoriées dans cette section, la liste de l'option `-o` peut également contenir la valeur littérale `name` afin de définir l'inclusion du nom de jeu de données dans la sortie.

L'exemple suivant illustre l'utilisation de la commande `zfs list` pour afficher le nom de jeu de données et des valeurs de propriété `sharenfs` et `mountpoint`.

```
# zfs list -r -o name,share.nfs,mountpoint users/home
NAME                          NFS       MOUNTPOINT
users/home                    on        /users/home
users/home/cindy              on        /users/home/cindy
users/home/gina               on        /users/home/gina
users/home/gina/projects      on        /users/home/gina/projects
users/home/gina/projects/fs1  on        /users/home/gina/projects/fs1
users/home/gina/projects/fs2  on        /users/home/gina/projects/fs2
users/home/mark               on        /users/home/mark
users/home/neil               on        /users/home/neil
```

Vous pouvez utiliser l'option `-t` pour spécifier le type de jeu de données à afficher. Le tableau suivant décrit les différents types valides.

TABLEAU 5-2 Types d'objets ZFS

Type	Description
`filesystem`	Systèmes de fichiers et clones

Type	Description
`volume`	Volumes
`share`	Partage de système de fichiers
`snapshot`	Clichés

Les options `-t` permettent de spécifier une liste séparée par des virgules des types de jeux de données à afficher. L'exemple suivant utilise l'utilisation simultanée des options `-t` et `-o` pour afficher le nom et la propriété `used` de l'ensemble de systèmes de fichiers :

```
# zfs list -r -t filesystem -o name,used users/home
NAME                           USED
users/home                    4.00G
users/home/cindy               548K
users/home/gina               2.00G
users/home/gina/projects      2.00G
users/home/gina/projects/fs1  1.00G
users/home/gina/projects/fs2  1.00G
users/home/mark               1.00G
users/home/neil               1.00G
```

Vous pouvez utiliser l'option `-H` pour omettre les en-tête `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 avec l'option `-H` : `zfs list`

```
# zfs list -r -H -o name users/home
users/home
users/home/cindy
users/home/gina
users/home/gina/projects
users/home/gina/projects/fs1
users/home/gina/projects/fs2
users/home/mark
users/home/neil
```

Gestion des propriétés ZFS

La gestion des propriétés de jeu de données s'effectue à l'aide des sous-commandes `set`, `inherit` et `get` de la commande `zfs`.

- “Paramétrage des propriétés ZFS” à la page 163
- “Héritage des propriétés ZFS” à la page 164
- “Interrogation des propriétés ZFS” à la page 165

Paramétrage 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 152.

La commande `zfs set` permet d'indiquer une séquence propriété/valeur au format *property=value*, suivie du nom du jeu de données. Vous ne pouvez définir ou modifier qu'une propriété à la fois au cours de chaque appel `zfs set`.

L'exemple suivant illustre la définition de la propriété `atime` sur la valeur `off` pour `tank/home`.

```
# zfs set atime=off tank/home
```

Vous pouvez également définir les propriétés des systèmes de fichiers une fois ces derniers créés. Par exemple :

```
# zfs create -o atime=off tank/home
```

Vous pouvez spécifier des valeurs de propriété numériques en utilisant les suffixes faciles à utiliser suivants (par taille croissante) : `BKMGTPEZ`. Ces suffixes peuvent être suivis de la lettre `b` (signifiant "byte", octet) à l'exception du suffixe `B`, qui fait déjà référence à cette unité de mesure. Les quatre invocations suivantes de `zfs set` sont des expressions numériques équivalentes qui définissent la propriété `quota` sur la valeur de 20 Go sur le système de fichiers `users/home/mark` :

```
# zfs set quota=20G users/home/mark
# zfs set quota=20g users/home/mark
# zfs set quota=20GB users/home/mark
# zfs set quota=20gb users/home/mark
```

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

```
# zfs set quota=20gb users/home/mark
cannot set property for '/users/home/mark': out of space
```

Les valeurs des propriétés non numériques respectent la casse et doivent être en lettres minuscules, à l'exception de `mountpoint`. Les valeurs de cette propriété peuvent contenir à la fois des majuscules et des minuscules.

Pour plus d'informations sur la commande `zfs set`, reportez-vous à la page de manuel `zfs`(1M).

Héritage des propriétés ZFS

A l'exception des quotas et des réservations, toutes les propriétés pouvant être définies héritent de la valeur du système de fichiers parent (sauf si un quota ou une réservation est explicitement défini sur le système de fichiers descendant). Si aucune valeur explicite n'est définie pour une propriété d'un système ascendant, la valeur par défaut de cette propriété est appliquée. Vous pouvez utiliser la commande `zfs inherit` pour effacer une valeur de la propriété et faire ainsi hériter la valeur du système de fichiers parent.

L'exemple suivant utilise la commande `zfs set` pour activer la compression du système de fichiers `tank/home/jeff`. La commande `zfs inherit` est ensuite exécutée afin de supprimer la valeur de la propriété `compression`, entraînant ainsi l'héritage de la valeur par défaut `off`. En effet, la propriété `compression` n'est définie localement ni pour `home`, ni pour `tank` ; la valeur par défaut est donc appliquée. Si la compression avait été activée pour ces deux systèmes, la valeur définie pour le système ascendant direct aurait été utilisée (en l'occurrence, `home`).

```
# zfs set compression=on tank/home/jeff
# zfs get -r compression tank/home
NAME                  PROPERTY     VALUE     SOURCE
tank/home             compression  off       default
tank/home/eric        compression  off       default
tank/home/eric@today  compression  -         -
tank/home/jeff        compression  on        local
# zfs inherit compression tank/home/jeff
# zfs get -r compression tank/home
NAME                  PROPERTY     VALUE     SOURCE
tank/home             compression  off       default
tank/home/eric        compression  off       default
tank/home/eric@today  compression  -         -
tank/home/jeff        compression  off       default
```

La sous-commande `inherit` est appliquée de manière récursive lorsque l'option `-r` est spécifiée. Dans l'exemple suivant, la commande entraîne l'héritage de la valeur de la propriété `compression` pour `tank/home` ainsi que pour ses éventuels descendants :

```
# zfs inherit -r compression tank/home
```

Remarque - L'utilisation de l'option `-r` supprime la valeur de propriété actuelle pour l'ensemble des systèmes de fichiers descendants.

Pour plus d'informations sur la commande `zfs inherit`, reportez-vous à la page de manuel `zfs`(1M).

Interrogation des 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 160. Toutefois, dans le cadre de requêtes complexes et pour les scripts, utilisez la commande `zfs get` afin de fournir des informations plus détaillées dans un format personnalisé.

La commande `zfs get` permet de récupérer les propriétés de jeu de données. L'exemple suivant illustre la récupération d'une seule valeur de propriété au sein d'un jeu de données :

```
# zfs get checksum tank/ws
NAME            PROPERTY       VALUE                     SOURCE
tank/ws         checksum       on                        default
```

La quatrième colonne `SOURCE` indique l'origine de la valeur de cette propriété. Le tableau ci-dessous définit les valeurs possibles de la source.

TABLEAU 5-3 Valeurs possibles de la colonne `SOURCE` (commande `zfs get`)

Valeur source	Description
`Valeur par défaut`	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.
`hérité de` *dataset-name*	La valeur de propriété est héritée du jeu de données parent spécifié par la chaîne *dataset-name*.
`Local`	La valeur de propriété a été définie de manière explicite pour ce jeu de données à l'aide de la commande `zfs set`.
`temporary`	Cette valeur de propriété a été définie à l'aide la commande `zfs mount` spécifiée avec l'option `-o` et n'est valide que pour la durée du montage. Pour plus d'informations sur les propriétés de point de montage temporaires, reportez-vous à la section “Utilisation de propriétés de montage temporaires” à la page 172.
None (Aucune)	Cette propriété est en lecture seule. Sa valeur est générée par ZFS.

Le mot-clé `all` permet de récupérer toutes les valeurs de propriétés du jeu de données. Les exemples suivants utilisent le mot-clé `all` :

```
# zfs get all tank/home
NAME       PROPERTY              VALUE                  SOURCE
tank/home  type                  filesystem             -
tank/home  creation              Mon Dec  3 13:10 2012  -
tank/home  used                  291K                   -
tank/home  available             58.7G                  -
tank/home  referenced            291K                   -
tank/home  compressratio         1.00x                  -
tank/home  mounted               yes                    -
tank/home  quota                 none                   default
```

```
tank/home  reservation           none                  default
tank/home  recordsize            128K                  default
tank/home  mountpoint            /tank/home            default
tank/home  sharenfs              off                   default
tank/home  checksum              on                    default
tank/home  compression           off                   default
tank/home  atime                 on                    default
tank/home  devices               on                    default
tank/home  exec                  on                    default
tank/home  setuid                on                    default
tank/home  readonly              off                   default
tank/home  zoned                 off                   default
tank/home  snapdir               hidden                default
tank/home  aclmode               discard               default
tank/home  aclinherit            restricted            default
tank/home  canmount              on                    default
tank/home  shareiscsi            off                   default
tank/home  xattr                 on                    default
tank/home  copies                1                     default
tank/home  version               5                     -
tank/home  utf8only              off                   -
tank/home  normalization         none                  -
tank/home  casesensitivity       mixed                 -
tank/home  vscan                 off                   default
tank/home  nbmand                off                   default
tank/home  sharesmb              off                   default
tank/home  refquota              none                  default
tank/home  refreservation        none                  default
tank/home  primarycache          all                   default
tank/home  secondarycache        all                   default
tank/home  usedbysnapshots       0                     -
tank/home  usedbydataset         291K                  -
tank/home  usedbychildren        0                     -
tank/home  usedbyrefreservation  0                     -
tank/home  logbias               latency               default
tank/home  sync                  standard              default
tank/home  rekeydate             -                     default
tank/home  rstchown              on                    default
```

L'option `-s` de `zfs get` vous permet de spécifier par type de source les propriétés à afficher. Cette option permet d'indiquer la liste des types de sources souhaités, séparés par une virgule. Seules les propriétés associées au type de source spécifié sont affichées. Les types de source valides sont `local`, `default`, `inherited`, `temporary` et `none`. L'exemple suivant indique l'ensemble des propriétés définies localement sur `tank/ws`.

```
# zfs get -s local all tank/ws
NAME     PROPERTY              VALUE                 SOURCE
tank/ws  compression           on                    local
```

Les options décrites ci-dessus peuvent être associées à l'option `-r` afin d'afficher de manière récursive les propriétés spécifiées sur les systèmes enfant du système de fichiers concerné. Dans l'exemple suivant, toutes les propriétés temporaires de l'ensemble des systèmes de fichiers de `tank/home` sont affichées de façon récursive :

```
# zfs get -r -s temporary all tank/home
NAME             PROPERTY       VALUE                      SOURCE
tank/home          atime          off                      temporary
tank/home/jeff     atime          off                      temporary
tank/home/mark     quota          20G                      temporary
```

Vous pouvez interroger les valeurs d'une propriété à l'aide de la commande `zfs get` sans spécifier le système de fichiers cible (la commande fonctionne sur tous les systèmes de fichiers et les pools). Par exemple :

```
# zfs get -s local all
tank/home               atime          off                    local
tank/home/jeff          atime          off                    local
tank/home/mark          quota          20G                    local
```

Pour plus d'informations sur la commande `zfs get`, reportez-vous à la page de manuel `zfs`(1M).

Envoi de requête sur les propriétés ZFS pour l'exécution de scripts

La commande `zfs get` prend en charge les options `-H` et `-o` qui permettent l'exécution de scripts. Vous pouvez utiliser l'option `-H` pour omettre les informations d'en-tête et pour remplacer un espace par un caractère de tabulation. L'uniformisation des espaces permet de faciliter l'analyse des données. Vous pouvez utiliser l'option-`o` pour personnaliser la sortie de l'une des manières 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 139.
- Une liste de champs littéraux séparés par des virgules (`name`, `value`, `property` et `source`) suivi d'un espace et d'un argument. En d'autres termes, il s'agit d'une liste de propriétés séparées par des virgules.

L'exemple suivant illustre la commande permettant de récupérer une seule valeur en spécifiant les options `-H` et `-o` de la commande `zfs get`:

```
# zfs get -H -o value compression tank/home
on
```

L'option `-p` indique les valeurs numériques sous leur forme exacte. Par exemple, 1 Mo serait signalé sous la forme 1000000. Cette option peut être utilisée comme suit :

```
# zfs get -H -o value -p used tank/home
182983742
```

L'option `-r` permet de récupérer de manière récursive les valeurs demandées pour l'ensemble des descendants et peut s'utiliser avec toutes les options mentionnées précédemment. Dans

l'exemple suivant, les options -H, -o et - r sont spécifiées afin de récupérer le nom du système de fichiers et la valeur de la propriété used pour export/home et ses descendants, tout en excluant les en-têtes dans la sortie :

```
# zfs get -H -o name,value -r used export/home
```

Montage de système de fichiers ZFS

Cette section décrit le processus de montage des systèmes de fichiers ZFS

Gestion des points de montage ZFS

Par défaut, un système de fichiers ZFS est automatiquement monté lors de sa création. Vous pouvez déterminer un comportement de point de montage spécifique pour un système de fichiers comme décrit dans cette section.

Vous pouvez également définir le point de montage par défaut du système de fichiers d'un pool lors de l'exécution de la commande de création zpool create en spécifiant l'option -m. Pour plus d'informations sur la création de pools de stockage, reportez-vous à la section "Création de pools de stockage ZFS" à la page 41.

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.

Vous pouvez remplacer le point de montage par défaut à l'aide de la commande zfs set pour définir la propriété mountpoint sur un chemin spécifique. ZFS crée automatiquement le point de montage spécifié, si nécessaire, et monte automatiquement le système de fichiers correspondant.

Les systèmes de fichiers ZFS sont automatiquement montés au moment de l'initialisation sans qu'il soit nécessaire d'éditer le fichier /etc/vfstab.

La propriété mountpoint est héritée. Par exemple, si le fichier pool/home est doté d'une propriété mountpoint définie sur /export/stuff, alors pool/home/user hérite de la valeur /export/stuff/user pour sa propriété mountpoint.

Afin d'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 à propos de `canmount`, reportez-vous à la section “Propriété canmount” à la page 153.

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 170.

Points de montage automatiques

- Lorsque vous modifiez la propriété `mountpoint` de `legacy` à `none` sur un chemin spécifique, le système de fichiers ZFS est automatiquement monté.
- Si le système de fichiers ZFS est géré automatiquement sans être monté et si la propriété `mountpoint` est modifiée, le système de fichiers reste démonté.

Les systèmes de fichiers dont la propriété `mountpoint` n'est pas définie sur `legacy` sont gérés par le système ZFS. L'exemple suivant illustre la création d'un système de fichiers dont le point de montage est automatiquement géré par le système ZFS :

```
# zfs create pool/filesystem
# zfs get mountpoint pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mountpoint    /pool/filesystem          default
# zfs get mounted pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mounted       yes                       -
```

Vous pouvez également définir la propriété `mountpoint` de manière explicite, comme dans l'exemple suivant :

```
# zfs set mountpoint=/mnt pool/filesystem
# zfs get mountpoint pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mountpoint    /mnt                      local
# zfs get mounted pool/filesystem
NAME             PROPERTY      VALUE                     SOURCE
pool/filesystem  mounted       yes                       -
```

Lorsque la propriété `mountpoint` est modifiée, le système de fichiers est automatiquement démonté de l'ancien point de montage et remonté sur le nouveau point de montage. Les répertoires de point de montage sont créés, le cas échéant. Si ZFS n'est pas en mesure de démonter un système de fichiers parce qu'il est actif, une erreur est signalée et un démontage manuel forcé doit être effectué.

Points de montage hérités

La gestion des systèmes de fichiers ZFS peut s'effectuer à l'aide d'outils hérités. Pour cela, la propriété `mountpoint` doit être définie sur `legacy`. Les systèmes de fichiers hérités sont alors gérés à l'aide des commandes `mount` et `umount` et du fichier `/etc/vfstab`. Lors de l'initialisation, le système de fichiers ZFS ne monte pas automatiquement les systèmes de fichiers hérités et les commandes ZFS `mount` et `umount` ne fonctionnent pas sur ces types de systèmes de fichiers. Les exemples suivants illustrent la définition et la gestion d'un système de fichiers ZFS hérité :

```
# zfs set mountpoint=legacy tank/home/eric
# mount -F zfs tank/home/eschrock /mnt
```

Pour monter automatiquement un système de fichiers hérité lors de l'initialisation, vous devez ajouter une entrée dans le fichier `/etc/vfstab`. L'exemple suivant montre l'entrée dans le fichier `/etc/vfstab` peut ressembler à ce qui suit :

```
#device         device        mount           FS      fsck    mount   mount
#to mount       to fsck       point           type    pass    at boot options
#

tank/home/eric  -  /mnt     zfs  -  yes  -
```

Les entrées `device to fsck` et `fsck pass` sont définies sur `-` car la commande `fsck` ne s'applique pas aux systèmes de fichiers ZFS. Pour plus d'informations sur l'intégrité des données ZFS, reportez-vous à la section “Sémantique transactionnelle” à la page 14.

Montage de système de fichiers ZFS

Le montage des systèmes de fichiers ZFS s'effectue automatiquement lors du processus de création ou lors de l'initialisation du système. Vous ne devez utiliser la commande `zfs mount` que lorsque vous devez modifier les options de montage ou monter/démonter explicitement les systèmes de fichiers.

Spécifiée sans argument, la commande `zfs mount` répertorie tous les systèmes de fichiers actuellement montés gérés par ZFS. Les points de montage hérités ne sont pas inclus. Par exemple :

```
# zfs mount | grep tank/home
zfs mount | grep tank/home
tank/home                       /tank/home
tank/home/jeff                  /tank/home/jeff
```

L'option `-a` permet de monter tous les systèmes de fichiers ZFS. Les systèmes de fichiers hérités ne sont pas montés. Par exemple :

```
# zfs mount -a
```

Par défaut, ZFS autorise uniquement le montage sur les s'agit pas d'un répertoire vide. Par exemple :

```
# zfs mount tank/home/lori
cannot mount 'tank/home/lori': filesystem already mounted
```

La gestion des points de montage hérités doit s'effectuer à l'aide des outils hérités. Toute tentative d'utilisation des outils ZFS génère une erreur. Par exemple :

```
# zfs mount tank/home/bill
cannot mount 'tank/home/bill': legacy mountpoint
use mount(1M) to mount this filesystem
# mount -F zfs tank/home/billm
```

Le montage d'un système de fichiers requiert l'utilisation d'un ensemble d'options basées sur les valeurs des propriétés associées au système de fichiers. Le tableau ci-dessous illustre la corrélation entre les propriétés et les options de montage :

TABLEAU 5-4 Options et propriétés de montage ZFS

Property	Option de montage
atime	atime/noatime
devices	devices/nodevices
exec	exec/noexec
nbmand	nbmand/nonbmand
readonly	ro/rw
setuid	setuid/nosetuid
xattr	xattr/noaxttr

L'option de montage `nosuid` représente un alias de `nodevices,nosetuid`.

Vous pouvez utiliser les fonctionnalités de montage en miroir NFSv4 pour faciliter la gestion des répertoires d'accueil ZFS montés via NFS.

Lorsque les systèmes de fichiers sont créés sur le serveur NFS, le client NFS peut les détecter automatiquement dans le montage existant d'un système de fichiers parent.

Par exemple, si le serveur `neo` partage déjà le système de fichiers `tank` et qu'il est monté sur le client `zee`, `/tank/baz` est automatiquement visible sur le client après avoir été créé sur le serveur.

```
zee# mount neo:/tank /mnt
zee# ls /mnt
baa    bar

neo# zfs create tank/baz

zee% ls /mnt
```

```
baa    bar    baz
zee% ls /mnt/baz
file1    file2
```

Utilisation de propriétés de montage temporaires

Si les options de montage décrites à la section précédente sont définies de manière explicite en spécifiant l'option -o avec la commande `zfs mount`, les valeurs des propriétés associées sont remplacées de manière temporaire. Ces valeurs de propriété sont désignées par la chaîne `temporary` dans la commande `zfs get` et reprennent leur valeur d'origine une fois le système de fichiers démonté. Si une valeur de propriété est modifiée alors que le système de fichiers est monté, la modification prend immédiatement effet et remplace toute valeur temporaire.

L'exemple suivant illustre la définition temporaire de l'option de montage en lecture seule sur le système de fichiers `tank/home/neil`. Le système de fichiers est censé être démonté.

```
# zfs mount -o ro users/home/neil
```

Pour modifier temporairement une valeur de propriété sur un système de fichiers monté, vous devez utiliser l'option spécifique `remount`. Dans l'exemple suivant, la propriété `atime` est temporairement définie sur la valeur `off` pour un système de fichiers monté :

```
# zfs mount -o remount,noatime users/home/neil
NAME             PROPERTY  VALUE  SOURCE
users/home/neil  atime     off    temporary
# zfs get atime users/home/perrin
```

Pour plus d'informations sur la commande `zfs mount`, reportez-vous à la page de manuel `zfs`(1M).

Démontage des systèmes de fichiers ZFS

Le démontage des systèmes de fichiers ZFS peut s'effectuer à l'aide de la commande `zfs unmount`. La commande `unmount`peut utiliser le point de montage ou le nom du système de fichiers en tant qu'argument.

L'exemple suivant illustre le démontage d'un système de fichiers avec l'argument de nom de système de fichiers :

```
# zfs unmount users/home/mark
```

L'exemple suivant illustre le démontage d'un système de fichiers avec l'argument de point de montage :

```
# zfs unmount /users/home/mark
```

Si le système de fichiers est occupé, la commande `unmount` échoue. L'option `-f` permet de forcer le démontage d'un système de fichiers. Le démontage forcé d'un système de fichiers requiert une attention particulière si le contenu de ce système est en cours d'utilisation. Ce type d'opération peut entraîner des comportements d'application imprévisibles.

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

Pour garantir la compatibilité ascendante, vous pouvez démonter les systèmes de fichiers ZFS à l'aide de la commande héritée `umount`. Par exemple :

```
# umount /tank/home/bob
```

Pour plus d'informations sur la commande `zfs unmount`, reportez-vous à la page de manuel `zfs`(1M).

Activation et annulation du partage des systèmes de fichiers ZFS

La version Oracle Solaris 11.1 simplifie l'administration de partage ZFS en utilisant l'héritage de propriété ZFS. La nouvelle syntaxe de partage est activée sur les pools exécutant la version de pool 34.

Vous trouverez ci-après pour le système de fichiers NFS et SMB packages, procédez comme suit :

- Client et les packages du serveur NFS
  - `service/file-system/nfs`(serveur)
  - `service/file-system/nfs` (client)

  Pour plus d'informations sur la configuration NFS, reportez-vous à " Gestion des systèmes de fichiers réseau dans Oracle Solaris 11.2 ".
- Client et les packages du serveur SMB
  - `service/file-system/smb`(serveur)
  - `service/file-system/smb` (client)

  Configuration pour plus d'informations, y compris la gestion des mots de passe SMB, reportez-vous à" Managing SMB Mounts in Your Local Environment " du manuel " Managing SMB File Sharing and Windows Interoperability in Oracle Solaris 11.2 ".

Il est possible de définir plusieurs partages par système de fichiers. Chaque partage est identifié par un nom unique. Vous pouvez définir les propriétés utilisées pour partager un chemin particulier dans un système de fichiers. Par défaut, aucun système de fichiers n'est partagé. En général, les services de serveur NFS ne sont pas démarrés tant qu'un partage n'est pas créé. Si

vous créez un partage valide, les services NFS sont démarrés automatiquement. Si la propriété `mountpoint` d'un système de fichiers ZFS est définie sur legacy, ce système de fichiers peut être partagé à l'aide de la commande héritée `share`.

- La propriété `share.nfs` remplace la propriété `sharenfs` des versions précédentes pour définir et publier un partage NFS.
- La propriété `share.smb` remplace la propriété `sharesmb` des versions précédentes pour définir et publier un partage SMB.
- Les propriétés `sharenfs` et `sharesmb` sont des alias des propriétés `share.nfs` et `sharenfs`.
- Le fichier `/etc/dfs/dfstab` n'est plus utilisé pour partager les systèmes de fichiers à l'initialisation du système. La configuration de ces propriétés partage les systèmes de fichiers automatiquement. SMF gère les informations de partage ZFS ou UFS de telle sorte que les systèmes de fichiers sont partagés automatiquement au redémarrage du système. Cette fonction signifie que tous les systèmes de fichiers dont la propriété `sharenfs` ou `sharesmb` n'est pas définie ne sont pas partagés à l'initialisation.
- L'interface `sharemgr` n'est plus disponible. La commande `share` héritée est toujours disponible pour créer un partage hérité. Voir les exemples ci-dessous.
- La commande `share -a` est similaire à la commande `share -ap` précédente et donc le système de fichier est persistant. L'option `share -p` n'est plus disponible.

Par exemple, si vous voulez partager le système de fichiers `tank/home`, utilisez une syntaxe similaire à la suivante :

```
# zfs set share.nfs=on tank/home
```

Dans l'exemple précédent, où la propriété `share.nfs` est définie sur le système de fichiers `tank/home`, la valeur de propriété `share.nfs` est héritée par tous les systèmes de fichiers descendants. Par exemple :

```
# zfs create tank/home/userA
# zfs create tank/home/userB
```

Vous pouvez également spécifier des valeurs de propriétés supplémentaires ou modifier des valeurs existantes sur des partages de système de fichiers existant. Par exemple :

```
# zfs set share.nfs.nosuid=on tank/home/userA
# zfs set share.nfs=on tank/home/userA
```

Syntaxe de partage ZFS héritée

La syntaxe d'Oracle Solaris 11 est toujours prise en charge et vous pouvez donc partager des systèmes de fichiers en deux étapes. Cette syntaxe est prise en charge dans toutes les versions de pool.

- Tout d'abord, utilisez la commande `zfs set share` pour créer un partage NFS ou SMB pour un système de fichiers ZFS.

```
# zfs create rpool/fs1
# zfs set share=name=fs1,path=/rpool/fs1,prot=nfs rpool/fs1
name=fs1,path=/rpool/fs1,prot=nfs
```

- Ensuite, activez la propriété `sharenfs` ou `sharesmb` pour publier le partage. Par exemple :

```
# zfs set sharenfs=on rpool/fs1
# grep fs1 /etc/dfs/sharetab
/rpool/fs1      fs1     nfs     sec=sys,rw
```

Les partages de système de fichiers peuvent être affichés grâce à la commande héritée `zfs get share`.

```
# zfs get share rpool/fs1
NAME       PROPERTY  VALUE  SOURCE
rpool/fs1  share     name=fs1,path=/rpool/fs1,prot=nfs  local
```

De plus, la commande `share`, pour partager un système de fichiers, similaire à la syntaxe d'Oracle Solaris 10, est toujours prise en charge pour partager tout répertoire au sein d'un système de fichiers. Par exemple, pour partager un système de fichiers ZFS :

```
# share -F nfs /tank/zfsfs
# grep zfsfs /etc/dfs/sharetab
/tank/zfsfs    tank_zfsfs     nfs     sec=sys,rw
```

La syntaxe ci-dessus équivaut à partager un système de fichiers UFS :

```
# share -F nfs /ufsfs
# grep ufsfs /etc/dfs/sharetab
/ufsfs          -               nfs     rw
/tank/zfsfs     tank_zfsfs      nfs     rw
```

Nouvelle syntaxe de partage ZFS

La commande `zfs set` permet de partager et de publier un système de fichiers ZFS via les protocoles NFS ou SMB. Vous pouvez également définir les propriétés `share.nfs` ou `share.smb` quand le système de fichiers est créé.

Par exemple, le système de fichiers `tank/sales` est créé et partagé. Les autorisations de partage en lecture et écriture s'applique par défaut à tout le monde. Le système de fichiers descendant `tank/sales/logs` est aussi partagé automatiquement, car la propriété `share.nfs` est héritée par les systèmes de fichiers descendants et l'accès du système de fichiers `tank/sales/log` est défini en lecture seule.

```
# zfs create -o share.nfs=on tank/sales
# zfs create -o share.nfs.ro=\* tank/sales/logs
# zfs get -r share.nfs tank/sales
NAME             PROPERTY   VALUE  SOURCE
tank/sales       share.nfs  on     local
```

```
tank/sales%      share.nfs  on     inherited from tank/sales
tank/sales/log   share.nfs  on     inherited from tank/sales
tank/sales/log%  share.nfs  on     inherited from tank/sales
```

Vous pouvez fournir un accès root à un système spécifique pour un système de fichiers partagé comme suit :

```
# zfs set share.nfs=on tank/home/data
# zfs set share.nfs.sec.default.root=neo.daleks.com tank/home/data
```

Partage ZFS avec héritage par propriété

Dans les pools mis à niveau à la version 34, une nouvelle syntaxe de partage simplifie la gestion des partages à l'aide de l'héritage de propriétés ZFS. Chaque caractéristique de partage devient une propriété `share` distincte. Les propriétés `share` sont identifiées par les noms comportant le préfixe `share.`. Des exemples de propriétés `share` sont `share.desc`, `share.nfs.nosuid`, et `share.smb.guestok`.

La propriété `share.nfs` contrôle l'activation du partage NFS. La propriété `share.smb` contrôle l'activation du partage SMB. Les noms des propriétés héritées `sharenfs` et `sharesmb` peuvent toujours être utilisés, car dans les nouveaux pools, `sharenfs` est un alias pour `share.nfs` et `sharesmb` est un alias pour `share.smb`. Si vous voulez partager le système de fichiers `tank/home`, utilisez une syntaxe similaire à la suivante :

```
# zfs set share.nfs=on tank/home
```

Dans cet exemple, la valeur de propriété `share.nfs` est héritée par tous les systèmes de fichiers descendants. Par exemple :

```
# zfs create tank/home/userA
# zfs create tank/home/userB
# grep tank/home /etc/dfs/sharetab
/tank/home      tank_home       nfs     sec=sys,rw
/tank/home/userA        tank_home_userA nfs     sec=sys,rw
/tank/home/userB        tank_home_userB nfs     sec=sys,rw
```

Héritage de partage ZFS dans les anciens pools

Dans les anciens pools, seules les propriétés `sharenfs` et `sharesmb` sont héritées par les systèmes de fichiers descendants. D'autres caractéristiques de partage sont stockées dans le fichier `.zfs/shares` pour chaque partage et ne sont pas héritées.

Une règle spéciale établit que quand un système de fichiers créé hérite de `sharenfs` ou `sharesmb` de son parent, un partage par défaut est créé pour ce système de fichiers à partir de la valeur `sharenfs` ou `sharesmb`. Notez que quand `sharenfs` est simplement activée, le partage par défaut créé dans un système de fichiers descendant possède uniquement les caractéristiques NFS par défaut. Par exemple :

```
# zpool get version tank
NAME  PROPERTY  VALUE  SOURCE
tank  version   33     default
# zfs create -o sharenfs=on tank/home
# zfs create tank/home/userA
# grep tank/home /etc/dfs/sharetab
/tank/home      tank_home       nfs     sec=sys,rw
/tank/home/userA        tank_home_userA nfs     sec=sys,r
```

Partages ZFS nommés

Vous pouvez aussi créer un partage *nommé*, qui offre plus de flexibilité dans la définition des autorisations et des propriétés dans un environnement SMB. Par exemple :

```
# zfs share -o share.smb=on tank/workspace%myshare
```

Dans l'exemple précédent, la commande `zfs share` crée un partage SMB nommé `myshare` du système de fichiers `tank/workspace`. Vous pouvez accéder au partage SMB et afficher ou définir des autorisations ou ACL spécifiques via le répertoire `.zfs/shares` du système de fichiers. Chaque partage SMB est représenté par un fichier `.zfs/shares` distinct. Par exemple :

```
# ls -lv /tank/workspace/.zfs/shares
-rwxrwxrwx+  1 root     root           0 May 15 10:31 myshare
0:everyone@:read_data/write_data/append_data/read_xattr/write_xattr
/execute/delete_child/read_attributes/write_attributes/delete
/read_acl/write_acl/write_owner/synchronize:allow
```

Les partages nommés héritent des propriétés de partage du système de fichiers parent. Si vous ajoutez la propriété `share.smb.guestok` au système de fichiers parent dans l'exemple précédent, la propriété est héritée par le partage nommé. Par exemple :

```
# zfs get -r share.smb.guestok tank/workspace
NAME                    PROPERTY           VALUE  SOURCE
tank/workspace          share.smb.guestok  on     inherited from tank
tank/workspace%myshare  share.smb.guestok  on     inherited from tank
```

Les partages nommés peuvent être utilisés dans l'environnement NFS lors de la définition de partages pour un sous-répertoire du système de fichiers. Par exemple :

```
# zfs create -o share.nfs=on -o share.nfs.anon=99 -o share.auto=off tank/home
# mkdir /tank/home/userA
# mkdir /tank/home/userB
# zfs share -o share.path=/tank/home/userA tank/home%userA
# zfs share -o share.path=/tank/home/userB tank/home%userB
# grep tank/home /etc/dfs/sharetab
/tank/home/userA        userA   nfs     anon=99,sec=sys,rw
/tank/home/userB        userB   nfs     anon=99,sec=sys,rw
```

L'exemple ci-dessus illustre que la désactivation de `share.auto` pour un système de fichiers désactive le partage automatique pour ce dernier tout en laissant toutes les autres propriétés

d'héritage intactes. Contrairement à la plupart des propriétés de partage, `share.auto` n'est pas héritable.

Les partages nommés servent également dans la création d'un partage NFS public. Un partage public peut uniquement être créé sur un partage NFS nommé. Par exemple :

```
# zfs create -o mountpoint=/pub tank/public
# zfs share -o share.nfs=on -o share.nfs.public=on tank/public%pubshare
# grep pub /etc/dfs/sharetab
/pub    pubshare        nfs     public,sec=sys,rw
```

Reportez-vous aux pages de manuel `share_nfs`(1M) et `share_smb`(1M) pour une description détaillée des propriétés de partages NFS et SMB.

Partages ZFS automatiques

Quand un partage automatique est créé, un nom de ressource unique est construit à partir du nom du système de fichiers. Le nom construit est une copie du nom du système de fichiers, à ceci près que les caractères du nom du système de fichiers qui ne sont pas autorisés dans un nom de ressource sont remplacés par des traits de soulignement (_). Par exemple, le nom de ressource de `data/home/john` est `data_home_john`.

La configuration d'un nom de propriété `share.autoname` permet de remplacer le nom de système de fichiers par un nom spécifique lors de la création du partage automatique. Le nom spécifique sert aussi à remplacer le nom de système de fichiers de préfixe en cas d'héritage. Par exemple :

```
# zfs create -o share.smb=on -o share.autoname=john data/home/john
# zfs create data/home/john/backups
# grep john /etc/dfs/sharetab
/data/home/john john    smb
/data/home/john/backups john_backups    smb
```

Si une commande `share` ou `zfs set share` est utilisée sur un système de fichiers qui n'a pas encore été partagé, sa valeur `share.auto` est automatiquement définie sur `off`. Les commandes héritées créent toujours des partages nommés. Cette règle spéciale empêche le partage automatique d'interférer avec le partage nommé qui est créé.

Affichage d'informations de partage ZFS

Affichez la valeur des propriétés de partage de fichier à l'aide de la commande `zfs get`. L'exemple suivant illustre comment afficher la propriété `share.nfs` pour un système de fichiers :

```
# zfs get share.nfs tank/sales
NAME        PROPERTY   VALUE  SOURCE
tank/sales  share.nfs  on     local
```

L'exemple suivant illustre comment afficher la propriété `share.nfs` pour des systèmes de fichiers descendants :

```
# zfs get -r share.nfs tank/sales
NAME             PROPERTY   VALUE  SOURCE
tank/sales       share.nfs  on     local
tank/sales%      share.nfs  on     inherited from tank/sales
tank/sales/log   share.nfs  on     inherited from tank/sales
tank/sales/log%  share.nfs  on     inherited from tank/sales
```

Les informations de propriétés de partage étendues ne sont pas disponibles dans la syntaxe de la commande `zfs get all`.

Vous pouvez afficher des détails spécifiques concernant des informations de partage NFS ou SMB à l'aide de la syntaxe suivante :

```
# zfs get share.nfs.all tank/sales
NAME        PROPERTY             VALUE  SOURCE
tank/sales  share.nfs.aclok      off    default
tank/sales  share.nfs.anon              default
tank/sales  share.nfs.charset.*  ...    default
tank/sales  share.nfs.cksum             default
tank/sales  share.nfs.index             default
tank/sales  share.nfs.log               default
tank/sales  share.nfs.noaclfab   off    default
tank/sales  share.nfs.nosub      off    default
tank/sales  share.nfs.nosuid     off    default
tank/sales  share.nfs.public     -      -
tank/sales  share.nfs.sec               default
tank/sales  share.nfs.sec.*      ...    default
```

Comme il existe de nombreuses propriétés de partage, envisagez de les afficher avec une valeur autre que par défaut. Par exemple :

```
# zfs get -e -s local,received,inherited share.all tank/home
NAME       PROPERTY           VALUE  SOURCE
tank/home  share.auto         off    local
tank/home  share.nfs          on     local
tank/home  share.nfs.anon     99     local
tank/home  share.protocols    nfs    local
tank/home  share.smb.guestok  on     inherited from tank
```

Modification des valeurs de propriété d'un partage ZFS

Vous pouvez modifier les valeurs de propriété d'un partage en spécifiant des propriétés nouvelles ou modifiées sur un partage de système de fichiers. Par exemple, si la propriété lecture seule est définie quand le système de fichiers est créé, la propriété ne peut pas être désactivée.

```
# zfs create -o share.nfs.ro=\* tank/data
# zfs get share.nfs.ro tank/data
```

```
NAME       PROPERTY              VALUE  SOURCE
tank/data  share.nfs.sec.sys.ro  *     local
# zfs set share.nfs.ro=none tank/data
# zfs get share.nfs.ro tank/data
NAME       PROPERTY              VALUE  SOURCE
tank/data  share.nfs.sec.sys.ro  off    local
```

Si vous créez un partage SMB, vous pouvez aussi ajouter le protocole de transfert NFS. Par exemple :

```
# zfs set share.smb=on tank/multifs
# zfs set share.nfs=on tank/multifs
# grep multifs /etc/dfs/sharetab
/tank/multifs   tank_multifs    nfs     sec=sys,rw
/tank/multifs   tank_multifs    smb     -
```

Supprimez le protocole SMB :

```
# zfs set share.smb=off tank/multifs
# grep multifs /etc/dfs/sharetab
/tank/multifs   tank_multifs    nfs     sec=sys,rw
```

Il est possible de renommer un partage nommé. Par exemple :

```
# zfs share -o share.smb=on tank/home/abc%abcshare
# grep abc /etc/dfs/sharetab
/tank/home/abc  abcshare        smb     -
# zfs rename tank/home/abc%abcshare tank/home/abc%a1share
# grep abc /etc/dfs/sharetab
/tank/home/abc  a1share        smb     -
```

Publication et annulation de publication de partages ZFS

Vous pouvez temporairement arrêter un partage nommé sans le détruire à l'aide de la commande `zfs unshare`. Par exemple :

```
# zfs unshare tank/home/abc%a1share
# grep abc /etc/dfs/sharetab
#
# zfs share tank/home/abc%a1share
# grep abc /etc/dfs/sharetab
/tank/home/abc  a1share smb     -
```

Quand la commande `zfs unshare` est utilisée, tous les partages de système de fichiers sont arrêtés. Ces partages restent ainsi jusqu'à ce que la commande `zfs share` soit émise pour le système de fichiers ou que les propriétés `share.nfs` ou `share.smb` soient définies pour le système de fichiers.

Les partages définis ne sont pas supprimés quand la commande `zfs unshare` est émise et sont de nouveau actifs à l'émission suivante de la commande `zfs share` pour le système de fichiers ou quand la propriété `share.nfs` ou `share.smb` est définie pour le système de fichiers.

Suppression d'un partage ZFS

Vous pouvez arrêter un partage de système de fichiers en définissant la propriété `share.nfs` ou `share.smb` sur off. Par exemple :

```
# zfs set share.nfs=off tank/multifs
# grep multifs /etc/dfs/sharetab
#
```

Vous pouvez supprimer définitivement un partage nommé avec la commande `zfs destroy`. Par exemple :

```
# zfs destroy tank/home/abc%a1share
```

Partage de fichiers ZFS au sein d'une zone non globale

Depuis Oracle Solaris 11, vous pouvez créer et publier des partages NFS dans une zone non globale d'Oracle Solaris.

- Lorsqu'un système de fichiers ZFS est monté et disponible dans une zone non globale, il peut être partagé dans cette zone.
- Un système de fichiers peut être partagé dans la zone globale à condition de ne pas être délégué à ou monté dans une zone non globale. Si un système de fichiers est ajouté à une zone non globale, il peut être partagé à l'aide de la commande héritée `share`.

Par exemple, les systèmes de fichiers `/export/home/data` et `export/home/data1` sont disponibles dans `zfszone`.

```
zfszone# share -F nfs /export/home/data
zfszone#  cat /etc/dfs/sharetab

zfszone# zfs set share.nfs=on tank/zones/export/home/data1
zfszone# cat /etc/dfs/sharetab
```

Problèmes de migration/transition de partage ZFS

Problèmes liés à la transition : reportez-vous aux points suivants

- **Importation de système de fichiers avec d'anciennes propriétés de partage** - Lors de l'importation d'un pool ou de la réception d'un flux de système de fichiers créé avant Oracle Solaris 11, les propriétés `sharenfs` et `sharesmb` incluent toutes les propriétés de partage directement dans la valeur de propriété. Dans la plupart des cas, ces propriétés de partage sont converties en un ensemble équivalent de partages nommés dès que chaque système de

fichiers est partagé. Etant donné que les opérations d'importation déclenchent un montage et un partage dans la plupart des cas, la conversion en partages nommés se produit directement pendant le processus d'importation.

- **Mise à niveau à partir Oracle Solaris 11** - Le premier partage de système de fichiers après une mise à niveau vers le pool version 34 peut prendre beaucoup de temps, car les partages nommés sont convertis au nouveau format. Les partages nommés créés par le processus de mise à niveau sont corrects mais ne peuvent pas profiter de l'héritage de propriétés de partage.
  - Afficher des valeurs de propriétés de partage :

    ```
    # zfs get share.nfs filesystem
    # zfs get share.smb filesystem
    ```
  - Si vous réinitialisez sur un ancien environnement d'initialisation, rétablissez les propriétés `sharenfs` et `sharesmb` à leur valeur d'origine.
- **Mise à niveau à partir d'Oracle Solaris 11** - Dans Oracle Solaris 11 et 11.1, les propriétés `sharenfs` et `sharesmb` peuvent uniquement avoir les valeurs `off` et `on`. Ces propriétés ne sont plus utilisées pour définir des caractéristiques de partage.

  Le fichier `/etc/dfs/dfstab` n'est plus utilisé pour partager les systèmes de fichiers à l'initialisation du système. Lors de l'initialisation, tous les systèmes de fichiers ZFS montés qui incluent des partages de système de fichiers sont automatiquement partagés. Un partage est activé quand `sharenfs` ou `sharesmb` sont définies sur `on`.

  L'interface `sharemgr` n'est plus disponible. La commande `share` héritée est toujours disponible pour créer un partage hérité. La commande `share -a` est similaire à la commande `share -ap` précédente et donc le système de fichier est persistant. L'option `share -p` n'est plus disponible.
- **Mise à niveau de votre système** : les partages ZFS seront incorrects si revenez à un environnement d'initialisation Oracle Solaris 11 antérieur en raison de modifications apportées aux propriétés dans cette version. Les partages non ZFS ne sont pas concernés. Si·vous·avez·l'intention·de·réinitialiser·un·ancien·environnement d'initialisation, enregistrez tout d'abord une copie de la configuration des partages existants avant l'opération `pkg update`, afin d'être en mesure de restaurer cette configuration sur les jeux de données ZFS.

  Dans l'environnement d'initialisation antérieur, utilisez la commande `sharemgr show -vp` pour répertorier tous les partages et leur configuration.

  Utilisez les commandes suivantes pour afficher les valeurs de propriété de partage :

  ```
  # zfs get sharenfs filesystem
  # zfs get sharesmb filesystem
  ```

  Si vous revenez à un ancien environnement de démarrage, réinitialisez les propriétés `sharenfs` et `sharesmb`, et tous les partages définis avec `sharemgr` à leurs valeurs d'origine.
- **Comportement hérité d'annulation de partage** : les commandes `unshare-a` ou `unshareall` permettent d'annuler le partage d'un système de fichiers, mais ne mettent pas

à jour le référentiel de partages SMF. Si vous tentez de republier le partage existant, les conflits sont recherchés dans le référentiel de partages et un message d'erreur s'affiche.

Dépannage des problèmes de partage de système de fichiers ZFS

Conditions reportez-vous aux points suivants : erreur de partage

- **Les partages nouveaux ou précédents ne sont pas partagé**
  - **Confirmer que les versions de pool et du système de fichiers sont à jour**Si les partages nouveaux ne sont pas partagés en définissant la propriété `share.nfs` ou `share.smb`, vérifiez que la version du pool est 34 et le système de fichiers 6.
  - **Le partage doit exister avant que les services NFS ne démarrent** : les services du serveur NFS ne sont pas exécutés tant que serveur un système de fichiers n'est pas partagé. D'abord créer le partage NFS et essayez ensuite d'y accéder le partage à distance.
  - **Un système avec des partages existants a été mis à niveau mais les partages ne sont pas disponibles** : un système avec les partages existants est mis à niveau, mais les tentatives de repartager échouent. Les partages risque de ne pas être partagés car la propriété `share.auto` est désactivée. Si la propriété `share.auto` est définie sur off, uniquement est partages nommés sont disponibles, ce qui impose la compatibilité avec la syntaxe de partage antérieure. Les partages existants, peuvent se présenter comme suit :

    ```
    # zfs get share
    NAME                      PROPERTY  VALUE  SOURCE
    tank/data                 share     name=data,path=/tank/data,prot=nfs  local
    ```

    1. Assurez-vous que la propriété `share.auto` est activée. Si ce n'est pas le cas, activez-le .

       ```
       # zfs get -r share.auto tank/data
       # zfs set share.auto=on tank/data
       ```

    2. Partagez à nouveau le système de fichiers.

       ```
       # zfs set -r share.nfs=on tank/data
       ```

    3. Il est possible que vous deviez supprimer les partages nommés puis les récréer pour que la commande précédente fonctionne.

       ```
       # zfs list -t share -Ho name -r tank/data | xargs -n1 zfs destroy
       ```

    4. Si nécessaire, recréez les partages nommés.

       ```
       # zfs create -o share.nfs=on tank/data%share
       ```

- **Les propriétés de partage incluant des partages nommés ne sont pas inclus dans les fichiers d'instantanés** : les propriétés et les fichiers `.zfs/shares` sont traités différemment

dans les opérations `zfs clone` et `zfs send`. Les fichiers `.zfs/shares` sont inclus dans des clichés et préservés dans les opérations `zfs clone` et `zfs send`. Pour une description du comportement des propriétés pendant les opérations `zfs send` et `zfs receive`, reportez-vous à la section “Application de différentes valeurs de propriété à un flux d'instantané ZFS” à la page 219. Après une opération de clonage, tous les fichiers proviennent du cliché pré-clonage, alors que les propriétés sont héritées de la nouvelle position du clone dans la hiérarchie du système de fichiers ZFS.

- **Une requête de partage nommé échoue** : Si une demande de création de partage nommé échoue car le partage serait en conflit avec le partage automatique, il peut s'avérer nécessaire de désactiver la propriété `auto.share`.
- **Un pool avec des partages a été exporté auparavant** : quand un pool est importé en lecture seule, ni ses propriétés et ses fichiers ne peuvent être modifiés, et la création d'un nouveau partage échoue. Si un partage était déjà établi avant l'exportation du pool, les caractéristiques de partage existantes sont utilisées, dans la mesure du possible.

Le tableau suivant identifie les états de partage connus et la manière de les résoudre, si nécessaire.

Etat de partage	Description	Résolution
`INVALID`	Le partage n'est pas valide car il est incohérent de manière interne ou il entre en conflit avec un autre partage.	Tentez de repartager le partage non valide à l'aide de la commande suivante : **`# zfs share FS%share`** L'utilisation de cette commande affiche une erreur concernant l'aspect du partage qui n'est pas validé. Corrigez cette erreur et retentez le partage.
`SHARED`	Le partage est partagé.	Pas nécessaire.
`UNSHARED`	Le partage est valide mais pas partagé.	Utilisez la commande `zfs share` pour repartager le partage en question ou le système de fichiers parent.
`UNVALIDATED`	Le partage n'est pas encore validé. Il se peut que le système de fichiers qui contient le partage ne soit pas dans un état partageable. Par exemple, il n'est pas monté ou est délégué à une zone autre que la zone actuelle. Autre possibilité, les propriétés FZS représentant le partage désiré ont été créées mais pas validées comme partage légal.	Utilisez la commande `zfs share` pour repartager le partage en question ou le système de fichiers parent. Si le système de fichiers est partageable, la tentative de repartager fonctionne (et l'état passe à shared) ou échoue (et l'état passe à invalid). Vous pouvez également utiliser la commande `share -A` pour répertorier tous les partages dans tous les systèmes de fichiers montés. Tous les partages des systèmes de fichiers montés sont alors résolus comme unshared (valides mais pas encore partagés) ou invalid.

Définition des quotas et réservations ZFS

La propriété `quota` permet de limiter la quantité d'espace disque disponible pour un système de fichiers. La propriété `reservation` permet quant à elle de garantir la disponibilité d'une certaine quantité d'espace disque pour un système de fichiers. Ces deux propriétés s'appliquent au système de fichiers sur lequel elles sont définies ainsi que tous les descendants de ce système de fichiers.

Par exemple, si un quota est défini pour le système de fichiers `tank/home`, la quantité totale d'espace disque utilisée par `tank/home`*et l'ensemble de ses descendants* ne peut pas excéder le quota défini. De même, si une réservation est définie pour le jeu de données `tank/home`, cette réservation s'applique à `tank/home` *et à tous ses descendants*. La quantité d'espace disque utilisée par un système de fichiers et par tous ses descendants est indiquée par la propriété `used`.

Les propriétés `refquota` et `refreservation` vous permettent de gérer l'espace d'un système de fichiers sans prendre en compte l'espace disque utilisé par les descendants, notamment les instantanés et les clones.

Dans cette version de Solaris, vous pouvez définir un quota d'utilisateur (*user*) ou de *groupe* sur la quantité d'espace disque utilisée par les fichiers appartenant à un utilisateur ou à un groupe spécifique. Les propriétés de quota d'utilisateur et de groupe ne peuvent pas être définies sur un volume, sur un système de fichiers antérieur à la version 4, ou sur un pool antérieur à la version 15.

Considérez les points suivants pour déterminer quelles fonctions de quota et de réservation conviennent le mieux à la gestion de vos systèmes de fichiers :

- Les propriétés `quota` et `reservation` conviennent à la gestion de l'espace disque utilisé par les systèmes de fichiers et leurs descendants.
- Les propriétés `refquota` et `refreservation` conviennent à la gestion de l'espace disque utilisé par les systèmes de fichiers.
- La définition d'une propriété `refquota` ou `refreservation` supérieure à une la propriété `quota` ou `reservation` n'a aucun effet. Lorsque vous définissez la propriété `quota` ou `refquota`, les opérations qui tentent de dépasser l'une de ces valeurs échouent. Il est possible de dépasser une valeur `quota` supérieure à une valeur `refquota`. Par exemple, si certains blocs d'instantanés sont modifiés, la valeur `quota` risque d'être dépassée avant la valeur `refquota`.
- Les quotas d'utilisateurs et de groupes permettent d'augmenter plus facilement l'espace disque contenant de nombreux comptes d'utilisateur, par exemple dans une université.

Pour plus d'informations sur la définition de quotas et réservations, reportez-vous aux sections “Définitions de quotas sur les systèmes de fichiers ZFS” à la page 186 et “Définition de réservations sur les systèmes de fichiers ZFS” à la page 189.

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 sur `tank/home/jeff` :

```
# zfs set quota=10G tank/home/jeff
# zfs get quota tank/home/jeff
NAME            PROPERTY  VALUE  SOURCE
tank/home/jeff  quota     10G    local
```

Les quotas affectent également la sortie des commandes `zfs list` et `df`. Par exemple :

```
# zfs list -r tank/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home          1.45M  66.9G    36K  /tank/home
tank/home/eric      547K  66.9G   547K  /tank/home/eric
tank/home/jeff      322K  10.0G   291K  /tank/home/jeff
tank/home/jeff/ws    31K  10.0G    31K  /tank/home/jeff/ws
tank/home/lori      547K  66.9G   547K  /tank/home/lori
tank/home/mark       31K  66.9G    31K  /tank/home/mark
# df -h /tank/home/jeff
Filesystem             Size  Used Avail Use% Mounted on
tank/home/jeff          10G  306K   10G   1% /tank/home/jeff
```

Toutefois, `tank/home` dispose de 66.9 GO d'espace disque, et `tank/home/jeff` et `tank/home/jeff/ws` disposent uniquement de 10 GO d'espace disponible, respectivement, en raison du quota défini pour `tank/home/jeff`.

Vous pouvez définir une propriété `refquota` sur un système de fichiers pour limiter l'espace disque occupé par le système de fichiers. Cette limite fixe ne comprend pas l'espace disque utilisé par les descendants. Par exemple, le quota de 10 Go de studentA n'est pas affecté par l'espace utilisé par les instantanés.

```
# zfs set refquota=10g students/studentA
# zfs list -t all -r students
NAME                          USED  AVAIL  REFER  MOUNTPOINT
students                      150M  66.8G    32K  /students
students/studentA             150M  9.85G   150M  /students/studentA
students/studentA@yesterday      0      -   150M  -
# zfs snapshot students/studentA@today
# zfs list -t all -r students
students                      150M  66.8G    32K  /students
students/studentA             150M  9.90G   100M  /students/studentA
students/studentA@yesterday  50.0M      -   150M  -
students/studentA@today          0      -   100M  -
```

Par souci de commodité, vous pouvez définir un autre quota pour un système de fichiers afin de vous aider à gérer l'espace disque utilisé par les instantanés. Par exemple :

```
# zfs set quota=20g students/studentA
# zfs list -t all -r students
NAME                          USED  AVAIL  REFER  MOUNTPOINT
students                      150M  66.8G    32K  /students
students/studentA             150M  9.90G   100M  /students/studentA
students/studentA@yesterday  50.0M      -   150M  -
students/studentA@today          0      -   100M  -
```

Dans ce scénario, `studentA` peut atteindre la limite maximale de `refquota` (10 Go), mais `studentA` peut supprimer des fichiers pour libérer de l'espace même en présence d'instantanés.

Dans l'exemple précédent, le plus petit des deux quotas (10 Go par rapport à 20 Go) s'affiche dans la sortie `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. Par exemple :

```
# zfs get refquota,quota students/studentA
NAME               PROPERTY  VALUE              SOURCE
students/studentA  refquota  10G                local
students/studentA  quota     20G                local
```

Définition de quotas d'utilisateurs et de groupes sur un système de fichiers ZFS

Vous pouvez définir un quota d'utilisateurs ou de groupes en utilisant respectivement les commandes `zfs userquota` et `zfs groupquota`. Par exemple :

```
# zfs create students/compsci
# zfs set userquota@student1=10G students/compsci
# zfs create students/labstaff
# zfs set groupquota@labstaff=20GB students/labstaff
```

Affichez le quota d'utilisateurs ou de groupes actuel comme suit :

```
# zfs get userquota@student1 students/compsci
NAME              PROPERTY            VALUE               SOURCE
students/compsci  userquota@student1  10G                 local
# zfs get groupquota@labstaff students/labstaff
NAME               PROPERTY             VALUE                SOURCE
students/labstaff  groupquota@labstaff  20G                  local
```

Vous pouvez afficher l'utilisation générale de l'espace disque par les utilisateurs et les groupes en interrogeant les propriétés suivantes :

```
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      350M   none
POSIX User  student1  426M    10G
# zfs groupspace students/labstaff
TYPE         NAME      USED  QUOTA
POSIX Group  labstaff  250M    20G
```

```
POSIX Group  root      350M   none
```

Pour identifier l'utilisation de l'espace disque d'un groupe ou d'un utilisateur, vous devez interroger les propriétés suivantes :

```
# zfs get userused@student1 students/compsci
NAME              PROPERTY           VALUE              SOURCE
students/compsci  userused@student1  550M               local
# zfs get groupused@labstaff students/labstaff
NAME               PROPERTY            VALUE               SOURCE
students/labstaff  groupused@labstaff  250                 local
```

Les propriétés de quota d'utilisateurs et de groupes ne sont pas affichées à l'aide de la commande `zfs get all`*dataset*, qui affiche une liste de toutes les autres propriétés du système de fichiers.

Vous pouvez supprimer un quota d'utilisateurs ou de groupes comme suit :

```
# zfs set userquota@student1=none students/compsci
# zfs set groupquota@labstaff=none students/labstaff
```

Les quotas d'utilisateurs et de groupes sur les systèmes de fichiers ZFS offrent les fonctionnalités suivantes :

- Un quota d'utilisateurs ou de groupes défini sur un système de fichiers parent n'est pas automatiquement hérité par un système de fichiers descendant.
- Cependant, le quota d'utilisateurs ou de groupes est appliqué lorsqu'un clone ou un instantané est créé à partir d'un système de fichiers lié à un quota d'utilisateurs ou de groupes. De même, un quota d'utilisateurs ou de groupes est inclus avec le système de fichiers lorsqu'un flux est créé à l'aide de la commande `zfs send`, même sans l'option `-R`.
- Les utilisateurs dénués de privilèges peuvent uniquement disposer de leur propre utilisation d'espace disque. L'utilisateur root ou l'utilisateur qui s'est vu accorder le privilège `userused` ou `groupused` peut accéder aux informations de comptabilité de l'espace disque utilisateur ou groupe de tout le monde.
- Les propriétés `userquota` et `groupquota` ne peuvent pas être définies sur les volumes ZFS, sur un système de fichiers antérieur à la version 4, ou sur un pool antérieur à la version 15.

L'application des quotas d'utilisateurs et de groupes peut être différée de quelques secondes. Ce délai signifie que les utilisateurs peuvent dépasser leurs quotas avant que le système ne le remarque et refuse d'autres écritures en affichant le message d'erreur `EDQUOT` .

Vous pouvez utiliser la commande `quota` héritée pour examiner les quotas d'utilisateurs dans un environnement NFS où un système de fichiers ZFS est monté, par exemple. Sans aucune option, la commande `quota` affiche uniquement la sortie en cas de dépassement du quota de l'utilisateur. Par exemple :

```
# zfs set userquota@student1=10m students/compsci
# zfs userspace students/compsci
```

```
TYPE        NAME      USED  QUOTA
POSIX User  root      350M   none
POSIX User  student1  550M    10M
# quota student1
Block limit reached on /students/compsci
```

Si vous réinitialisez le quota d'utilisateurs et que la limite du quota n'est plus dépassée, vous devez utiliser la commande `quota -v` pour examiner le quota de l'utilisateur. Par exemple :

```
# zfs set userquota@student1=10GB students/compsci
# zfs userspace students/compsci
TYPE        NAME      USED  QUOTA
POSIX User  root      350M   none
POSIX User  student1  550M    10G
# quota student1
# quota -v student1
Disk quotas for student1 (uid 102):
Filesystem     usage  quota  limit    timeleft  files  quota  limit    timeleft
/students/compsci
563287 10485760 10485760           -      -      -      -           -
```

Définition de réservations sur les systèmes de fichiers ZFS

Une *réservation* ZFS désigne une quantité d'espace disque du pool garantie pour un jeu de données. Dès lors, pour réserver une quantité d'espace disque pour un jeu de données, cette quantité doit être actuellement disponible sur le pool. La quantité totale d'espace non utilisé des réservations ne peut pas dépasser la quantité d'espace disque non utilisé du pool. La définition et l'affichage des réservations ZFS s'effectuent respectivement à l'aide des commandes `zfs set` et `zfs get`. Par exemple :

```
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
NAME            PROPERTY     VALUE   SOURCE
tank/home/bill  reservation  5G      local
```

Les réservations peuvent affecter la sortie de la commande `zfs list`. Par exemple :

```
# zfs list -r tank/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home          5.00G  61.9G    37K  /tank/home
tank/home/bill       31K  66.9G    31K  /tank/home/bill
tank/home/jeff      337K  10.0G   306K  /tank/home/jeff
tank/home/lori      547K  61.9G   547K  /tank/home/lori
tank/home/mark       31K  61.9G    31K  /tank/home/mark
```

Notez que `tank/home` utilise 5 Go d'espace bien que la quantité totale d'espace à laquelle `tank/home` et ses descendants font référence est bien inférieure à 5 Go. L'espace utilisé reflète l'espace

réservé pour `tank/home/bill`. Les réservations sont prises en compte dans le calcul de l'espace disque utilisé du système de fichiers parent et non dans le quota, la réservation ou les deux.

```
# zfs set quota=5G pool/filesystem
# zfs set reservation=10G pool/filesystem/user1
cannot set reservation for 'pool/filesystem/user1': size is greater than
available space
```

Un jeu de données peut utiliser davantage d'espace disque que sa réservation, du moment que le pool dispose d'un espace non réservé et disponible et que l'utilisation actuelle du jeu de données se trouve en dessous des quotas. Un jeu de données ne peut pas utiliser un espace disque réservé à un autre jeu de données.

Les réservations ne sont pas cumulatives. En d'autres termes, l'exécution d'une nouvelle commande `zfs set` pour un jeu de données déjà associé à une réservation n'entraîne pas l'ajout de la nouvelle réservation à la réservation existante. La seconde réservation remplace la première. Par exemple :

```
# zfs set reservation=10G tank/home/bill
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
NAME            PROPERTY     VALUE   SOURCE
tank/home/bill  reservation  5G      local
```

Vous pouvez définir une réservation `refreservation` pour garantir un espace disque ne contenant aucun instantané ou clone au jeu de données. Cette valeur est prise en compte dans le calcul de l'espace utilisé des jeux de données parent et vient en déduction des quotas et réservations des jeux de données parent. Par exemple :

```
# zfs set refreservation=10g profs/prof1
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
profs                    10.0G  23.2G    19K  /profs
profs/prof1                10G  33.2G    18K  /profs/prof1
```

Vous pouvez également définir une valeur de réservation pour le même jeu de données afin de garantir l'espace du jeu de données et pas de l'espace des instantanés. Par exemple :

```
# zfs set reservation=20g profs/prof1
# zfs list
NAME                      USED  AVAIL  REFER  MOUNTPOINT
profs                    20.0G  13.2G    19K  /profs
profs/prof1                10G  33.2G    18K  /profs/prof1
```

Les réservations régulières sont prises en compte dans le calcul de l'espace utilisé du parent.

Dans l'exemple précédent, le plus petit des deux quotas (10 Go par rapport à 20 Go) s'affiche dans la sortie `zfs list`. Pour afficher la valeur des deux quotas, utilisez la commande `zfs get`. Par exemple :

```
# zfs get reservation,refreserv profs/prof1
NAME         PROPERTY        VALUE        SOURCE
```

```
profs/prof1  reservation     20G          local
profs/prof1  refreservation  10G          local
```

Lorsque la propriété `refreservation` est définie, un instantané n'est autorisé que si suffisamment d'espace non réservé est disponible dans le pool au-delà de cette réservation afin de pouvoir contenir le nombre actuel d'octets *référencés* dans le jeu de données.

Chiffrement des systèmes de fichiers ZFS

Le chiffrement se définit comme le processus de codage de données à des fins de confidentialité ; le propriétaire des données nécessite une clé pour pouvoir accéder aux données codées. Les avantages de l'utilisation du chiffrement ZFS sont les suivants :

- Le chiffrement ZFS est intégré à l'ensemble des commandes ZFS. A l'instar d'autres opérations ZFS, les opérations de chiffrement telles que la modification et le renouvellement de clés sont effectuées en ligne.
- Vous pouvez utiliser vos pools de stockage existants pour autant qu'ils aient été mis à niveau. Vous avez la possibilité de chiffrer des systèmes de fichiers spécifiques.
- Les données sont chiffrées à l'aide de la norme AES (Advanced Encryption Standard, Norme de chiffrement avancé) avec des longueurs de clé de 128, 192 et 256 dans les modes de fonctionnement CCM et GCM.
- Le chiffrement ZFS utilise la structure cryptographique Oracle Solaris, ce qui lui donne automatiquement accès aux éventuelles accélérations matérielles et implémentations logicielles optimisées disponibles des algorithmes de chiffrement.
- A l'heure actuelle, vous ne pouvez pas chiffrer un système de fichiers root ZFS ou autres composants de système d'exploitation, tels que le répertoire `/var`, même s'il s'agit d'un système de fichiers distinct.
- Le chiffrement ZFS peut être transmis aux systèmes de fichiers descendants.
- Un utilisateur standard peut créer un système de fichiers chiffré et gérer les activités clés si les droits d'accès `create`, `mount`, `keysource`, `checksum` et `encryption` lui ont été attribués.

Vous pouvez définir une stratégie de chiffrement lors de la création d'un système de fichiers ZFS, mais cette stratégie ne peut pas être modifiée. Par exemple, la propriété de chiffrement est activée lors de la création du système de fichiers `tank/home/darren`. La stratégie de chiffrement par défaut consiste à demander une phrase de passe comptant au moins 8 caractères.

```
# zfs create -o encryption=on tank/home/darren
Enter passphrase for 'tank/home/darren': xxxxxxx
Enter again: xxxxxxxx
```

Assurez-vous que le chiffrement est activé sur le système de fichiers. Par exemple :

```
# zfs get encryption tank/home/darren
NAME              PROPERTY    VALUE        SOURCE
tank/home/darren  encryption  on           local
```

L'algorithme de chiffrement par défaut est `aes-128-ccm` lorsque la valeur de chiffrement d'un système de fichiers est `on`.

Une *clé d'encapsulation* est utilisée pour chiffrer les clés de chiffrement effectives des données. La clé d'encapsulation est transmise de la commande `zfs` au noyau, comme dans l'exemple ci-dessus au moment de la création du système de fichiers chiffré. Une clé d'encapsulation peut se trouver dans un fichier (au format raw ou hexadécimal) ou être dérivée d'une phrase de passe.

Le format et l'emplacement de la clé d'encapsulation sont spécifiés dans la propriété `keysource` de la manière suivante :

```
keysource=format,location
```

- Formats possibles :
  - `raw` : octets bruts de la clé
  - `hex` : chaîne de clé hexadécimale
  - `passphrase` : chaîne de caractères générant une clé
- Emplacements possibles :
  - `prompt` – vous êtes invité à saisir une clé ou une phrase de passe lorsque le système de fichiers est créé ou monté
  - `file:///`*filename* – Emplacement dans le système de fichiers de la clé ou phrase de passe.
  - `pkcs11` – URI décrivant l'emplacement d'une clé ou d'une phrase de passe dans un jeton PKCS#11
  - `https://`*location* – Emplacement sur le serveur sécurisé de la clé ou phrase de passe. Le transport d'informations de clé non chiffrées à l'aide de cette méthode n'est pas recommandé. Un `GET` sur l'URL retourne seulement la valeur de clé ou la phrase de passe, en fonction de ce qui a été demandé dans la partie format de la propriété `keysource`.

    Lors de l'utilisation d'un localisateur `https://` pour la `keysource`, le certificat que ce serveur présente doit être sécurisé par `libcurl` et OpenSSL. Ajoutez votre propre ancre sécurisée ou certificat autosigné au magasin de certificats dans `/etc/openssl/certs`. Placez le certificat de format PEM dans le répertoire `/etc/certs/CA` et exécutez la commande suivante :

    ```
    # svcadm refresh ca-certificates
    ```

Si le format spécifié par `keysource` est *passphrase*, la clé d'encapsulation est dérivée de la passe de phrase. Dans le cas contraire, la valeur de la propriété `keysource` pointe vers la clé d'encapsulation effective, sous forme d'octets bruts ou au format hexadécimal. Vous pouvez indiquer que la phrase de passe doit être stockée dans un fichier ou dans un flux d'octets bruts que l'utilisateur est invité à saisir, ce qui n'est probablement adapté qu'à l'écriture de scripts.

Lorsque les valeurs de la propriété `keysource` d'un système de fichiers correspondent à `passphrase`, la clé d'encapsulation est dérivée de la phrase de passe à l'aide de `PKCS#5 PBKD2`

et d'un salt généré de façon aléatoire pour chaque système de fichiers. Cela signifie que la même phrase de passe génère une clé d'encapsulation différente lorsqu'elle est utilisée sur des systèmes de fichiers descendants.

Les systèmes de fichiers descendants héritent de la stratégie de chiffrement du système de fichiers parent, et celle-ci ne peut pas être supprimée. Par exemple :

```
# zfs snapshot tank/home/darren@now
# zfs clone tank/home/darren@now tank/home/darren-new
Enter passphrase for 'tank/home/darren-new': xxxxxxx
Enter again: xxxxxxxx
# zfs set encryption=off tank/home/darren-new
cannot set property for 'tank/home/darren-new': 'encryption' is readonly
```

Si vous devez copier ou migrer des systèmes de fichiers ZFS chiffrés ou non chiffrés, tenez compte des points suivants :

- A l'heure actuelle, vous ne pouvez pas envoyer un flux de jeu de données non chiffré et le recevoir en tant que flux chiffré, même si le chiffrement est activé sur le jeu de données du pool de réception.
- Vous pouvez utiliser les commandes suivantes pour migrer des données non chiffrées vers un pool ou système de données où le chiffrement est activé :
  - `cp -r`
  - `find | cpio`
  - `tar`
  - `rsync`
- Un flux de système de fichiers chiffré répliqué peut être reçu sur un système de fichiers chiffré et les données restent chiffrées. Pour plus d'informations, reportez-vous à l'Exemple 5-4, “Envoi et réception d'un système de fichiers ZFS chiffré”.

Modification des clés d'un système de fichiers ZFS chiffré

Vous pouvez modifier la clé d'encapsulation d'un système de fichiers chiffré à l'aide de la commande `zfs key -c`. La clé d'encapsulation existante doit avoir été chargée auparavant : soit lors de l'initialisation, soit par chargement explicite de la clé du système de fichiers (`zfs key -l`), soit par montage du système de fichiers (`zfs mount` *filesystem*). Par exemple :

```
# zfs key -c tank/home/darren
Enter new passphrase for 'tank/home/darren': xxxxxxxx
Enter again: xxxxxxxx
```

Dans l'exemple suivant, la clé d'encapsulation est modifiée et la valeur de la propriété `keysource` est modifiée pour indiquer que la clé d'encapsulation provient d'un fichier.

```
# zfs key -c -o keysource=raw,file:///media/stick/key tank/home/darren
```

La clé de chiffrement des données d'un système de fichiers chiffré peut être modifiée à l'aide de la commande `zfs key -K`, mais la nouvelle clé de chiffrement est uniquement utilisée pour les nouvelles données écrites. Cette fonctionnalité peut être utilisée pour assurer la conformité avec les directives NIST 800-57 relatives à la limitation dans le temps de la clé de chiffrement de données. Par exemple :

```
# zfs key -K tank/home/darren
```

Dans l'exemple ci-dessus, la clé de chiffrement des données n'est ni visible ni directement gérée par vous. En outre, la délégation `keychange` est requise pour effectuer une opération de modification de clé.

Les algorithmes de chiffrement suivants sont disponibles :

- `aes-128-ccm`, `aes-192-ccm`, `aes-256-ccm`
- `aes-128-gcm`, `aes-192-gcm`, `aes-256-gcm`

La propriété `keysource` ZFS identifie le format et l'emplacement de la clé qui encapsule les clés de chiffrement des données du système de fichiers. Par exemple :

```
# zfs get keysource tank/home/darren
NAME              PROPERTY   VALUE              SOURCE
tank/home/darren  keysource  passphrase,prompt  local
```

La propriété `rekeydate` ZFS identifie la date de la dernière opération `zfs key -K`. Par exemple :

```
# zfs get rekeydate tank/home/darren
NAME              PROPERTY   VALUE                  SOURCE
tank/home/darren  rekeydate  Wed Jul 25 16:54 2012  local
```

Si les propriétés `creation` et `rekeydate` d'un système de fichiers chiffré possèdent la même valeur, cela signifie que le système de fichiers n'a jamais fait l'objet d'un renouvellement de clés via une opération `zfs key-K`.

Gestion des clés de chiffrement ZFS

Les clés de chiffrement ZFS peuvent être gérées de différentes manières, selon vos besoins, sur le système local ou à distance, si un emplacement centralisé est nécessaire.

- **Localement** – Les exemples ci-dessus montrent que la clé d'encapsulation peut être une invite de phrase de passe ou une clé brute stockée dans un fichier du système local.
- **A distance** – Les informations de clé peuvent être stockées à distance en utilisant un système de gestion des clés centralisé comme Oracle Key Manager ou un service Web qui prend en charge une simple demande GET sur un URI http ou https. Les informations de clé d'Oracle Key Manager sont accessibles à un système Oracle Solaris à l'aide d'un jeton PKCS#11.

Pour plus d'informations sur la gestion des clés de chiffrement ZFS : http://www.oracle.com/technetwork/articles/servers-storage-admin/manage-zfs-encryption-1715034.html

Pour plus d'informations sur l'utilisation d'Oracle Key Manager pour gérer les informations de clé :

http://docs.oracle.com/cd/E24472_02/

Autorisations de délégation d'opérations sur les clés ZFS

Passez en revue les descriptions d'autorisations relatives à la délégation d'opérations sur les clés suivantes :

- Le chargement et le déchargement d'un système de fichiers à l'aide des commandes `zfs key -l` et `zfs key -u` requièrent l'autorisation `key`. Dans la plupart des cas, l'autorisation `mount` est également requise.
- La modification de la clé d'un système de fichiers à l'aide des commandes `zfs key-c` et `zfs key-K` requiert l'autorisation `keychange`.

Envisagez de déléguer des autorisations distinctes pour l'utilisation (chargement ou déchargement) et la modification de clés, de manière à mettre en place un modèle de gestion des clés à deux personnes. Par exemple, déterminez les utilisateurs qui pourront utiliser les clés et ceux qui seront autorisés à les modifier. Vous pouvez aussi spécifier que toute modification de clé requiert la présence des deux utilisateurs. Ce modèle vous permet également de bâtir un système de dépôt de clé (key escrow).

Montage d'un système de fichiers ZFS chiffré

Tenez compte des points suivants lorsque vous tentez de monter un système de fichiers ZFS chiffré :

- Si une clé de système de fichiers chiffré n'est pas disponible lors de l'initialisation, le système de fichiers n'est pas monté automatiquement. Par exemple, un système de fichiers dont la stratégie de chiffrement est définie sur `passphrase,prompt` ne sera pas monté lors de l'initialisation car le processus d'initialisation ne s'interrompra pas pour afficher une invite de saisie de phrase de passe.
- Si vous souhaitez monter un système de fichiers avec une stratégie de chiffrement définie sur `passphrase,prompt` lors de l'initialisation, vous devez explicitement le monter à l'aide de la commande `zfs mount` et spécifier la phrase de passe ou utiliser la commande `zfs key -l` pour être invité à saisir la clé après l'initialisation du système.

  Par exemple :

```
# zfs mount -a
Enter passphrase for 'tank/home/darren': xxxxxxxx
```

```
Enter passphrase for 'tank/home/ws': xxxxxxxx
Enter passphrase for 'tank/home/mark': xxxxxxxx
```

- Si la propriété `keysource` d'un système de fichiers chiffré pointe vers un fichier appartenant à un autre système de fichiers, l'ordre de montage des systèmes de fichiers peut déterminer si le système de fichiers est monté lors de l'initialisation ou non, en particulier si le fichier est placé sur un média amovible.

Mise à niveau des systèmes de fichiers ZFS chiffrés

Avant de mettre à niveau un système Solaris 11 vers Solaris 11.1, veillez à ce que les systèmes de fichiers chiffrés soient montés. Effectuez le montage des systèmes de fichiers chiffrés et fournissez les phrases de passe si vous y êtes invité.

```
# zfs mount -a
Enter passphrase for 'pond/amy': xxxxxxxx
Enter passphrase for 'pond/rory': xxxxxxxx
# zfs mount | grep pond
pond                            /pond
pond/amy                        /pond/amy
pond/rory                       /pond/rory
```

Effectuez ensuite une mise à niveau des systèmes de fichiers chiffrés.

```
# zfs upgrade -a
```

Si vous tentez de mettre à niveau des systèmes de fichiers ZFS chiffrés qui ne sont pas montés, un message semblable à celui-ci s'affiche :

```
# zfs upgrade -a
cannot set property for 'pond/amy': key not present
```

En outre, la sortie de `zpool status` affiche parfois des données endommagées.

```
# zpool status -v pond
.
.
.
pond/amy:<0x1>
pond/rory:<0x1>
```

Si les erreurs mentionnées ci-dessus se produisent, effectuez un remontage des systèmes de fichiers chiffrés comme indiqué ci-dessus. Ensuite, effacez et nettoyez les erreurs de pool.

```
# zpool scrub pond
# zpool clear pond
```

Pour plus d'informations sur la mise à niveau des systèmes de fichiers, reportez-vous à “Mise à niveau des systèmes de fichiers ZFS” à la page 202.

Interactions entre les propriétés de compression, de suppression des doublons et de chiffrement ZFS

Tenez compte des points suivants lorsque vous utilisez les propriétés de compression, de suppression des doublons et de chiffrement ZFS.

- Lorsqu'un fichier est écrit, les données sont compressées, chiffrées et la somme de contrôle est vérifiée. Lorsque cela est possible, les données sont ensuite dédupliquées.
- Lorsqu'un fichier est lu, la somme de contrôle est vérifiée et les données sont déchiffrées. Si nécessaire, les données sont ensuite décompressées.
- Si la propriété `dedup` est activée sur un système de fichiers chiffré qui est également cloné et si les commandes `zfs key-K` ou `zfs clone-K` n'ont pas été utilisées sur les clones, les données de tous les clones sont dédupliquées, lorsque cela est possible.

Exemples de chiffrement de systèmes de fichiers ZFS

EXEMPLE 5-1 Chiffrement d'un système de fichiers ZFS à l'aide d'une clé raw

Dans l'exemple suivant, une clé de chiffrement `aes-256-ccm` est générée à l'aide de la commande `pktool` et écrite dans un fichier, `/cindykey.file`.

```
# pktool genkey keystore=file outkey=/cindykey.file keytype=aes keylen=256
```

Le fichier `/cindykey.file` est ensuite spécifié lorsque le système de fichiers `tank/home/cindy` est créé.

```
# zfs create -o encryption=aes-256-ccm -o keysource=raw,file:///cindykey.file
tank/home/cindy
```

EXEMPLE 5-2 Chiffrement d'un système de fichiers ZFS à l'aide d'un autre algorithme de chiffrement

Vous pouvez créer un pool de stockage ZFS et faire en sorte que tous les systèmes de fichiers du pool de stockage héritent d'un algorithme de chiffrement. Dans l'exemple qui suit, le pool `users` est créé et le système de fichiers `users/home` est créé et chiffré à l'aide d'une phrase de passe. L'algorithme de chiffrement par défaut est `aes-128-ccm`.

Le système de fichiers `users/home/mark` est ensuite créé et chiffré à l'aide de l'algorithme de chiffrement `aes-256-ccm`.

```
# zpool create -O encryption=on users mirror c0t1d0 c1t1d0 mirror c2t1d0 c3t1d0
Enter passphrase for 'users': xxxxxxxx
Enter again: xxxxxxxx
# zfs create users/home
# zfs get encryption users/home
NAME        PROPERTY    VALUE        SOURCE
users/home  encryption  on           inherited from users
# zfs create -o encryption=aes-256-ccm users/home/mark
# zfs get encryption users/home/mark
NAME               PROPERTY    VALUE        SOURCE
users/home/mark    encryption  aes-256-ccm  local
```

EXEMPLE 5-3 Clonage d'un système de fichiers ZFS chiffré

Si le système de fichiers clone hérite de la propriété `keysource` du même système de fichiers que son instantané d'origine, une nouvelle propriété `keysource` n'est pas nécessaire, et vous n'êtes pas invité à saisir une nouvelle phrase de passe lorsque `keysource=passphrase,prompt` . La même propriété `keysource` est utilisée pour le clone. Par exemple :

Par défaut, vous n'êtes pas invité à saisir une clé lors du clonage d'un descendant d'un système de fichiers chiffré.

```
# zfs create -o encryption=on tank/ws
Enter passphrase for 'tank/ws': xxxxxxxx
Enter again: xxxxxxxx
# zfs create tank/ws/fs1
# zfs snapshot tank/ws/fs1@snap1
# zfs clone tank/ws/fs1@snap1 tank/ws/fs1clone
```

Si vous souhaitez créer une nouvelle clé pour le système de fichiers clone, utilisez la commande `zfs clone-K`.

Si vous clonez un système de fichiers chiffré et non un système de fichiers chiffré descendant, vous êtes invité à fournir une nouvelle clé. Par exemple :

```
# zfs create -o encryption=on tank/ws
Enter passphrase for 'tank/ws': xxxxxxxx
Enter again: xxxxxxxx
# zfs snapshot tank/ws@1
# zfs clone tank/ws@1 tank/ws1clone
Enter passphrase for 'tank/ws1clone': xxxxxxxx
Enter again: xxxxxxxx
```

EXEMPLE 5-4 Envoi et réception d'un système de fichiers ZFS chiffré

Dans l'exemple suivant, l'instantané `tank/home/darren@snap1` est créé à partir du système de fichiers chiffré `/tank/home/darren`. Ensuite, l'instantané est envoyé vers `bpool/snaps` avec la propriété de chiffrement activée, si bien que les données résultantes reçues sont chiffrées. Toutefois, le flux `tank/home/darren@snap1` n'est pas chiffré pendant le processus d'envoi.

```
# zfs get encryption tank/home/darren
NAME              PROPERTY    VALUE        SOURCE
tank/home/darren  encryption  on           local
# zfs snapshot tank/home/darren@snap1
# zfs get encryption bpool/snaps
NAME         PROPERTY    VALUE        SOURCE
bpool/snaps  encryption  on           inherited from bpool
# zfs send tank/home/darren@snap1 | zfs receive bpool/snaps/darren1012
# zfs get encryption bpool/snaps/darren1012
NAME                    PROPERTY    VALUE        SOURCE
bpool/snaps/darren1012  encryption  on           inherited from bpool
```

Dans ce cas, une nouvelle clé est automatiquement générée pour le système de fichiers chiffré reçu.

Migration de systèmes de fichiers ZFS

Vous pouvez utiliser la fonctionnalité de migration shadow pour migrer des systèmes de fichiers comme suit :

- Système de fichiers ZFS local ou distant vers système de fichiers ZFS cible
- Système de fichiers UFS local ou distant vers système de fichiers ZFS cible

La *migration shadow* est un processus permettant d'extraire les données à migrer :

- Créez un système de fichiers ZFS vide.
- Définissez la propriété `shadow` sur un système de fichiers ZFS vide, qui constitue le système de fichiers cible (ou shadow), de manière à ce qu'elle pointe vers le système de fichiers à migrer.
- Les données du système de fichiers à migrer sont copiées vers le système de fichiers shadow.

Vous pouvez identifier le système de fichiers à migrer à l'aide de l'URI de la propriété shadow de l'une des manières suivantes :

- `shadow=file:///`*path* : utilisez cette syntaxe pour migrer un système de fichiers local
- `shadow=nfs://`*host/path* –: utilisez cette syntaxe pour migrer un système de fichiers NFS

Tenez compte des points suivants lors de la migration de systèmes de fichiers :

- Le système de fichiers à migrer doit être défini sur lecture seule. Si le système de fichiers n'est pas défini sur lecture seule, les modifications en cours risquent de ne pas être migrées.
- Le système de fichiers cible doit être totalement vide.
- Si le système est réinitialisé pendant la migration, la migration se poursuit une fois l'initialisation terminée.

- Le contenu d'un répertoire ou d'un fichier dont la migration n'est pas terminée est inaccessible jusqu'à ce que l'ensemble du contenu ait été migré.
- Si vous souhaitez migrer les informations relatives aux UID, GID et ACL vers le système de fichiers shadow d'une migration NFS, assurez-vous que les informations du service de noms sont accessibles entre le système local et le système distant. Vous pouvez envisager de copier un sous-ensemble des données du système de fichiers à migrer afin de tester la migration et de vous assurer que toutes les informations sont correctement migrées avant d'effectuer une migration de grande envergure via NFS.
- La migration des données d'un système de fichiers via NFS peut être lente, selon la bande passante du réseau. Soyez patient.
- La commande `shadowstat` permet de surveiller la migration d'un système de fichiers et fournit les données suivantes :
  - La colonne `BYTES XFRD` indique le nombre d'octets transférés au système de fichiers shadow.
  - La colonne `BYTES LEFT` est sans cesse mise à jour jusqu'à ce que la migration soit presque terminée. ZFS n'identifie pas la quantité de données à migrer au début de la migration car ce processus peut être trop long.
  - Pensez à utiliser les informations `BYTES XFRD` et `ELAPSED TIME` pour estimer la durée du processus de migration.

▼ Migration d'un système de fichiers vers un système de fichiers ZFS

1. **Si vous migrez des données à partir d'un serveur NFS distant, assurez-vous que les informations du service de noms sont accessibles sur les deux systèmes.**

   Pour des migrations de grande envergure à l'aide de NFS, envisagez d'effectuer une migration test d'un sous ensemble de données afin de vous assurer que les informations relatives aux UID, GUID, et ACL sont correctement migrées.

2. **Installez le package de migration shadow sur le système où les données doivent être migrées, si nécessaire, et activez le service `shadowd` pour faciliter le processus de migration.**

   ```
   # pkg install shadow-migration

   # svcadm enable shadowd
   ```

   Si vous n'activez pas le processus `shadowd`, vous devrez restaurer le réglage `none` de la propriété `shadow` à l'issue du processus de migration.

3. **Définissez le système de fichiers local ou distant à migrer sur lecture seule.**

   Si vous migrez un système de fichiers ZFS local, définissez-le sur lecture seule. Par exemple :

```
# zfs set readonly=on tank/home/data
```

Si vous migrez un système de fichiers distant, partagez-le en lecture seule. Par exemple :

```
# share -F nfs -o ro /export/home/ufsdata
# share
-               /export/home/ufsdata   ro   ""
```

4. **Créez un nouveau système de fichiers ZFS et définissez la propriété shadow de celui-ci sur le système de fichiers à migrer.**

   Par exemple, si vous migrez un système de fichiers ZFS local, `rpool/old`, vers un nouveau système de fichiers ZFS, `users/home/shadow`, définissez la propriété `shadow` sur `rpool/old` lors de la création du système de fichiers `users/home/shadow`.

   ```
   # zfs create -o shadow=file:///rpool/old users/home/shadow
   ```

   Par exemple, pour migrer `/export/home/ufsdata` à partir d'un serveur distant, définissez la propriété `shadow` lors de la création du système de fichiers ZFS.

   ```
   # zfs create -o shadow=nfs://neo/export/home/ufsdata users/home/shadow2
   ```

5. **Vérifiez la progression de la migration.**

   Par exemple :

   ```
   # shadowstat
   EST
   BYTES   BYTES           ELAPSED
   DATASET                         XFRD    LEFT    ERRORS  TIME
   users/home/shadow               45.5M   2.75M   -       00:02:31
   users/home/shadow               55.8M   -       -       00:02:41
   users/home/shadow               69.7M   -       -       00:02:51
   No migrations in progress
   ```

   Lorsque la migration est terminée, la propriété `shadow` est définie sur `none`.

   ```
   #zfs get -r shadow users/home/shadow*
   NAME                PROPERTY  VALUE   SOURCE
   users/home/shadow   shadow    none    -
   users/home/shadow2  shadow    none    -
   ```

Dépannage des migrations de systèmes de fichiers ZFS

Tenez compte des points suivants lors du dépannage des problèmes de migration ZFS :

- Si le système de fichiers à migrer n'est pas défini sur lecture seule, certaines données ne sont pas migrées.

- Si le système de fichiers cible n'est pas vide lorsque la propriété `shadow` est définie, la migration des données ne se lance pas.
- Si vous ajoutez ou supprimez des données du système de fichiers à migrer alors que la migration est en cours, ces modifications risquent de ne pas être migrées.
- Si vous tentez de modifier le montage du système de fichiers shadow alors que la migration est en cours, le message suivant s'affiche :

```
# zfs set mountpoint=/users/home/data users/home/shadow3
cannot unmount '/users/home/shadow3': Device busy
```

Mise à niveau des systèmes de fichiers ZFS

Si vous possédez des systèmes de fichiers ZFS d'une version antérieure de Solaris, vous pouvez procéder à la mise à niveau de vos systèmes de fichiers à l'aide de la commande `zfs upgrade` afin de tirer parti des fonctions du système de fichiers dans la version actuelle. De plus, cette commande vous avertit lorsque vos systèmes de fichiers exécutent des versions antérieures.

Par exemple, la version de ce système de fichiers est la version 5 actuelle.

```
# zfs upgrade
This system is currently running ZFS filesystem version 5.

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

Utilisez cette commande pour identifier les fonctions disponibles pour chaque version des systèmes de fichiers.

```
# zfs upgrade -v
The following filesystem versions are supported:

VER  DESCRIPTION
---  --------------------------------------------------------
1   Initial ZFS filesystem version
2   Enhanced directory entries
3   Case insensitive and File system unique identifier (FUID)
4   userquota, groupquota properties
5   System attributes

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

Pour plus d'informations sur la mise à niveau des systèmes de fichiers chiffrés, reportez-vous à “Mise à niveau des systèmes de fichiers ZFS chiffrés” à la page 196.

♦ ♦ ♦ CHAPITRE 6

Utilisation des instantanés et des clones ZFS Oracle Solaris

Ce chapitre fournit des informations sur la création et la gestion d'instantanés et de clones ZFS Oracle Solaris. Des informations concernant l'enregistrement des instantanés sont également fournies.

Ce chapitre contient les sections suivantes :

Présentation des instantanés ZFS

Un *instantané* est une copie en lecture seule d'un système de fichiers ou d'un volume. La création des instantanés est quasiment immédiate. Initialement, elle ne consomme pas d'espace disque supplémentaire au sein du pool. Toutefois, à mesure que les données contenues dans le jeu de données actif changent, l'instantané consomme de l'espace disque en continuant à faire référence aux anciennes données et empêche donc la libération de l'espace disque.

Les instantanés ZFS présentent les caractéristiques suivantes :

- Persistance au cours des réinitialisations de système
- Théoriquement, le nombre maximal d'instantanés est de 2^{64} instantanés.
- Les instantanés n'utilisent aucune sauvegarde de secours distincte. Les instantanés consomment de l'espace disque provenant directement du pool de stockage auquel appartient le système de fichiers ou le volume à partir duquel ils ont été créés.

- Une seule opération, dite atomique, permet de créer rapidement des instantanés récursifs. Ceux-ci sont tous créés simultanément ou ne sont pas créés du tout. Grâce à ce type d'opération d'instantané atomique, une certaine cohérence des données d'instantané est assurée, y compris pour les systèmes de fichiers descendants.

Il n'est pas possible d'accéder directement aux instantanés de volumes, mais ils peuvent être clonés, sauvegardés, restaurés, etc. Pour plus d'informations sur la sauvegarde d'un instantané ZFS, reportez-vous à la section “Envoi et réception de données ZFS” à la page 213.

Création et destruction d'instantanés ZFS

Les instantanés sont créés à l'aide de la commande `zfs snapshot` ou `zfs snap`, qui accepte comme unique argument 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 18.

Dans l'exemple suivant, un instantané de `tank/home/cindy` nommé `friday`est créé.

```
# zfs snapshot tank/home/cindy@friday
```

Vous pouvez créer des instantanés pour tous les systèmes de fichiers descendants à l'aide de l'option `-r`. Par exemple :

```
# zfs snapshot -r tank/home@snap1
# zfs list -t snapshot -r tank/home
NAME                   USED  AVAIL  REFER  MOUNTPOINT
tank/home@snap1           0      -  2.11G  -
tank/home/cindy@snap1     0      -   115M  -
tank/home/lori@snap1      0      -  2.00G  -
tank/home/mark@snap1      0      -  2.00G  -
tank/home/tim@snap1       0      -  57.3M  -
```

Les propriétés des instantanés ne sont pas modifiables. Les propriétés des jeux de données ne peuvent pas être appliquées à un instantané. Par exemple :

```
# zfs set compression=on tank/home/cindy@friday
cannot set property for 'tank/home/cindy@friday':
this property can not be modified for snapshots
```

La commande `zfs destroy` permet de détruire les instantanés. Par exemple :

```
# zfs destroy tank/home/cindy@friday
```

La destruction d'un jeu de données est impossible s'il existe des instantanés du jeu de données. Par exemple :

```
# zfs destroy tank/home/cindy
cannot destroy 'tank/home/cindy': filesystem has children
use '-r' to destroy the following datasets:
tank/home/cindy@tuesday
tank/home/cindy@wednesday
tank/home/cindy@thursday
```

En outre, si des clones ont été créés à partir d'un instantané, ils doivent être détruits avant que l'instantané ne puisse être détruit.

Pour plus d'informations sur la sous-commande `destroy`, reportez-vous à la section “Destruction d'un système de fichiers ZFS” à la page 137.

Conservation des clichés ZFS

Ou différentes stratégies de conservation des données instantané automatique peut se traduire par les instantanés les plus anciens sont par exemple détruit. Si la suppression d'un fait partie d'un instantané zfs send et une opération de réception en cours, l'opération risque d'échouer. Pour éviter ce scénario, placez éventuellement un blocage à un instantané.

La *conservation* d'un instantané empêche sa destruction. En outre, cette fonction permet de supprimer un instantané contenant des clones, en attendant la suppression du dernier clone à l'aide de la commande `zfs destroy-d`. Chaque instantané est associé à un décompte de référence utilisateur initialisé sur 0 (zéro). Ce nombre augmente de 1 à chaque fois qu'un instantané est conservé et diminue de 1 à chaque fois qu'un instantané conservé est libéré.

Dans la version précédente d'Oracle Solaris, les instantanés pouvaient uniquement être détruits à l'aide de la commande `zfs destroy` s'ils ne contenaient aucun clone. Dans cette version d'Oracle Solaris, les instantanés doivent également renvoyer un décompte de référence utilisateur égal à 0 (zéro).

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

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

Vous pouvez utiliser l'option `-r` pour conserver récursivement les instantanés de tous les systèmes de fichiers descendants. Par exemple :

```
# zfs snapshot -r tank/home@now
# zfs hold -r keep tank/home@now
```

Cette syntaxe permet d'ajouter une référence `keep` unique à cet instantané ou à ce jeu d'instantanés. Chaque instantané possède son propre espace de noms de balise dans lequel

chaque balise de conservation doit être unique. Si un instantané est conservé, les tentatives de destruction de ce dernier à l'aide de la commande `zfs destroy` échoueront. Par exemple :

```
# zfs destroy tank/home/cindy@snap1
cannot destroy 'tank/home/cindy@snap1': dataset is busy
```

Pour détruire un instantané conservé, utilisez l'option `-d`. Par exemple :

```
# zfs destroy -d tank/home/cindy@snap1
```

Utilisez la commande `zfs holds` pour afficher la liste des instantanés conservés. Par exemple :

```
# zfs holds tank/home@now
NAME           TAG   TIMESTAMP
tank/home@now  keep  Fri Aug  3 15:15:53 2012

# zfs holds -r tank/home@now
NAME                 TAG   TIMESTAMP
tank/home/cindy@now  keep  Fri Aug  3 15:15:53 2012
tank/home/lori@now   keep  Fri Aug  3 15:15:53 2012
tank/home/mark@now   keep  Fri Aug  3 15:15:53 2012
tank/home/tim@now    keep  Fri Aug  3 15:15:53 2012
tank/home@now        keep  Fri Aug  3 15:15:53 2012
```

Vous pouvez utiliser la commande `zfs release` pour libérer un instantané ou un jeu d'instantanés conservé. Par exemple :

```
# zfs release -r keep tank/home@now
```

Si l'instantané est libéré, l'instantané peut être détruit à l'aide de la commande `zfs destroy`. Par exemple :

```
# zfs destroy -r tank/home@now
```

Deux nouvelles propriétés permettent d'identifier les informations de conservation d'un instantané :

- La propriété `defer_destroy` est définie sur `on` si l'instantané a été marqué en vue d'une destruction différée à l'aide de la commande `zfs destroy -d`. Dans le cas contraire, la propriété est définie sur `off`.
- La propriété `userrefs` également appelée décompte de *référence utilisateur*, est définie sur le nombre de conservations pour cet instantané.

Renommage d'instantanés ZFS

Vous pouvez renommer les instantanés. Cependant, ils doivent rester dans le même pool et dans le même jeu de données dans lequel il ont été créés. Par exemple :

```
# zfs rename tank/home/cindy@snap1 tank/home/cindy@today
```

En outre, la syntaxe de raccourci suivante est équivalente à la syntaxe précédente :

```
# zfs rename tank/home/cindy@snap1 today
```

L'opération de renommage (`rename`) d'instantané n'est pas prise en charge, car le nom du pool cible et celui du système de fichiers ne correspondent pas au pool et au système de fichiers dans lesquels l'instantané a été créé :

```
# zfs rename tank/home/cindy@today pool/home/cindy@saturday
cannot rename to 'pool/home/cindy@today': snapshots must be part of same
dataset
```

Vous pouvez renommer de manière récursive les instantanés à l'aide de la commande `zfs rename- r`. Par exemple :

```
# zfs list -t snapshot -r users/home
NAME                        USED  AVAIL  REFER  MOUNTPOINT
users/home@now             23.5K      -  35.5K  -
users/home@yesterday           0      -    38K  -
users/home/lori@yesterday      0      -  2.00G  -
users/home/mark@yesterday      0      -  1.00G  -
users/home/neil@yesterday      0      -  2.00G  -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -t snapshot -r users/home
NAME                        USED  AVAIL  REFER  MOUNTPOINT
users/home@now             23.5K      -  35.5K  -
users/home@2daysago            0      -    38K  -
users/home/lori@2daysago       0      -  2.00G  -
users/home/mark@2daysago       0      -  1.00G  -
users/home/neil@2daysago       0      -  2.00G  -
```

Affichage et accès des instantanés ZFS

Par défaut, les instantanés ne sont plus affichés dans la sortie `zfs list`. Vous devez utiliser la commande `zfs list -t snapshot` pour afficher les informations relatives aux instantanés. Ou activez la propriété de pool `listsnapshots`. Par exemple :

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

Les instantanés des systèmes de fichiers sont accessibles dans le répertoire `.zfs/snapshot` au sein du root du système de fichiers. Par exemple, si `tank/home/cindy` est monté sur `/home/cindy`, les données d'instantané`tank/home/cindy@thursday` sont accessibles dans le répertoire `/home/cindy/.zfs/snapshot/thursday`.

```
# ls /tank/home/cindy/.zfs/snapshot
thursday   tuesday    wednesday
```

Vous pouvez répertorier les instantanés comme suit :

```
# zfs list -t snapshot -r tank/home
NAME                       USED  AVAIL  REFER  MOUNTPOINT
tank/home/cindy@tuesday     45K      -  2.11G  -
tank/home/cindy@wednesday   45K      -  2.11G  -
tank/home/cindy@thursday      0      -  2.17G  -
```

Vous pouvez répertorier les instantanés qui ont été créés pour un système de fichiers particulier comme suit :

```
# zfs list -r -t snapshot -o name,creation tank/home
NAME                       CREATION
tank/home/cindy@tuesday    Fri Aug  3 15:18 2012
tank/home/cindy@wednesday  Fri Aug  3 15:19 2012
tank/home/cindy@thursday   Fri Aug  3 15:19 2012
tank/home/lori@today       Fri Aug  3 15:24 2012
tank/home/mark@today       Fri Aug  3 15:24 2012
```

Comptabilisation de l'espace disque des instantanés ZFS

Lors de la création d'un instantané, son espace disque est initialement partagé entre l'instantané et le système de fichiers et éventuellement avec des instantanés précédents. Lorsque le système de fichiers change, l'espace disque précédemment partagé devient dédié à l'instantané, et il est compté dans la propriété `used` de l'instantané. De plus, la suppression d'instantanés peut augmenter la quantité d'espace disque dédié à d'autres instantanés (et, par conséquent, *utilisé* par ceux-ci).

La valeur de la propriété `referenced` de l'espace d'instantané est identique à la valeur du système de fichiers au moment où l'instantané a été créé.

Vous pouvez identifier des informations supplémentaires sur la façon dont les valeurs de la propriété `used` sont utilisées. Les nouvelles propriétés de système de fichiers en lecture seule décrivent l'utilisation de l'espace disque pour les clones, les systèmes de fichiers et les volumes. Par exemple :

```
$ zfs list -o space -r rpool
NAME                     AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
rpool                     124G  9.57G         0    302K              0      9.57G
rpool/ROOT                124G  3.38G         0     31K              0      3.38G
rpool/ROOT/solaris        124G  20.5K         0       0              0      20.5K
rpool/ROOT/solaris/var    124G  20.5K         0   20.5K              0          0
rpool/ROOT/solaris-1      124G  3.38G     66.3M   3.14G              0       184M
rpool/ROOT/solaris-1/var  124G   184M     49.9M    134M              0          0
rpool/VARSHARE            124G  39.5K         0   39.5K              0          0
rpool/dump                124G  4.12G         0   4.00G           129M          0
rpool/export              124G    63K         0     32K              0        31K
rpool/export/home         124G    31K         0     31K              0          0
```

```
rpool/swap                124G  2.06G         0   2.00G          64.7M          0
```

Pour une description de ces propriétés, reportez-vous au Tableau 5-1, “Description des propriétés ZFS natives”.

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/cindy` est annulé (rollback) jusqu'à son instantané `tuesday` :

```
# zfs rollback tank/home/cindy@tuesday
cannot rollback to 'tank/home/cindy@tuesday': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
tank/home/cindy@wednesday
tank/home/cindy@thursday
# zfs rollback -r tank/home/cindy@tuesday
```

Dans cet exemple, les instantanés `wednesday` et `thursday` sont détruits en raison de la restauration de l'instantané `tuesday` précédent.

```
# zfs list -r -t snapshot -o name,creation tank/home/cindy
NAME                      CREATION
tank/home/cindy@tuesday  Fri Aug  3 15:18 2012
```

Identification des différences entre des instantanés ZFS (`zfs diff`)

Vous pouvez déterminer les différences entre des instantanés ZFS en utilisant la commande `zfs diff`.

Supposons par exemple que les deux instantanés suivants sont créés :

```
$ ls /tank/home/tim
fileA
$ zfs snapshot tank/home/tim@snap1
$ ls /tank/home/tim
fileA  fileB
$ zfs snapshot tank/home/tim@snap2
```

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

```
$ zfs diff tank/home/tim@snap1 tank/home/tim@snap2
M       /tank/home/tim/
+       /tank/home/tim/fileB
```

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

Dans la sortie suivante, le `R` indique qu'un fichier dans un instantané a été renommé.

```
$ mv /tank/cindy/fileB /tank/cindy/fileC
$ zfs snapshot tank/cindy@snap2
$ zfs diff tank/cindy@snap1 tank/cindy@snap2
M       /tank/cindy/
R       /tank/cindy/fileB -> /tank/cindy/fileC
```

Le tableau suivant résume les modifications apportées au fichier ou au répertoire identifiées par la commande `zfs diff`.

Modification de répertoire ou de fichier	Identifiant
Le fichier ou le répertoire a été modifié ou le lien d'un répertoire ou d'un fichier a changé	`M`
Le fichier ou le répertoire est présent dans l'ancien instantané mais pas dans le plus récent	–
Le fichier ou le répertoire est présent dans l'instantané le plus récent mais pas dans le plus ancien.	`+`
Le fichier ou le répertoire a été renommé	`R`

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

Présentation des clones ZFS

Un *clone* est un volume ou un système de fichiers accessible en écriture et dont le contenu initial est similaire à celui du jeu de données à partir duquel il a été créé. Tout comme pour les

instantanés, la création d'un clone est quasiment instantanée et ne consomme initialement aucun espace disque supplémentaire. En outre, vous pouvez prendre un instantané d'un clone.

Les clones se créent uniquement à partir d'un instantané. Lors du clonage d'un instantané, une dépendance implicite se crée entre le clone et l'instantané. Bien que le clone soit créé à une autre emplacement dans la hiérarchie de système de fichiers, l'instantané d'origine ne peut pas être supprimé tant que le clone existe. La propriété `origin` indique cette dépendance et la commande `zfs destroy` répertorie ces dépendances, le cas échéant.

Un clone n'hérite pas des propriétés du jeu de données à partir duquel il a été créé. Les commandes `zfs get` et `zfs set` permettent d'afficher et de modifier les propriétés d'un jeu de données cloné. Pour plus d'informations sur la configuration des propriétés de jeux de données ZFS, reportez-vous à la section “Paramétrage des propriétés ZFS” à la page 163.

Dans la mesure où un clone partage initialement son espace disque avec l'instantané d'origine, la valeur de la propriété `used` est initialement égale à zéro. A mesure que le clone est modifié, il utilise de plus en plus d'espace disque. La propriété `used` de l'instantané d'origine ne tient pas compte de l'espace disque consommé par le clone.

Création d'un clone ZFS

Pour créer un clone, utilisez la commande `zfs clone` en spécifiant l'instantané à partir duquel créer le clone, ainsi que le nom du nouveau volume ou système de fichiers. Le nouveau volume ou système de fichiers peut se trouver à tout emplacement de la hiérarchie ZFS. Le nouveau jeu de données est du même type (un système de fichiers ou un volume, par exemple) que celui de l'instantané à partir duquel le clone a été créé. Vous ne pouvez pas créer le clone d'un système de fichiers dans un autre pool que celui de l'instantané du système de fichiers d'origine.

Dans l'exemple suivant, un nouveau clone appelé `tank/home/matt/bug123` possédant le même contenu initial que l'instantané `tank/ws/gate@yesterday` est créé :

```
# zfs snapshot tank/ws/gate@yesterday
# zfs clone tank/ws/gate@yesterday tank/home/matt/bug123
```

Dans l'exemple suivant, un espace de travail est créé à partir de l'instantané `projects/newproject@today` pour un utilisateur temporaire, sous le nom `projects/teamA/tempuser`. Ensuite, les propriétés sont configurées dans l'espace de travail cloné.

```
# zfs snapshot projects/newproject@today
# zfs clone projects/newproject@today projects/teamA/tempuser
# zfs set share.nfs=on projects/teamA/tempuser
# zfs set quota=5G projects/teamA/tempuser
```

Destruction d'un clone ZFS

La commande `zfs destroy` permet de détruire les clones ZFS. Par exemple :

```
# zfs destroy tank/home/matt/bug123
```

Les clones doivent être détruits préalablement à la destruction de l'instantané parent.

Remplacement d'un système de fichiers ZFS par un clone ZFS

La commande `zfs promote` permet de remplacer un système de fichiers ZFS actif par un clone de ce système de fichiers. Cette fonction facilite le clonage et le remplacement des systèmes de fichiers pour que le système de fichiers *original* devienne le clone du système de fichiers spécifié. En outre, cette fonction permet de détruire le système de fichiers à partir duquel le clone a été créé. Il est impossible de détruire un système de fichiers d'origine possédant des clones actifs, sans le remplacer par l'un de ses clones. Pour plus d'informations sur la destruction des clones, reportez-vous à “Destruction d'un clone ZFS” à la page 212.

Dans l'exemple suivant, le système de fichiers `tank/test/productA` est cloné, puis le clone du système de fichiers (`tank/test/productAbeta`) devient le système de fichiers `tank/test/productA` d'origine.

```
# zfs create tank/test
# zfs create tank/test/productA
# zfs snapshot tank/test/productA@today
# zfs clone tank/test/productA@today tank/test/productAbeta
# zfs list -r tank/test
NAME                       USED  AVAIL  REFER  MOUNTPOINT
tank/test                  104M  66.2G    23K  /tank/test
tank/test/productA         104M  66.2G   104M  /tank/test/productA
tank/test/productA@today      0      -   104M  -
tank/test/productAbeta        0  66.2G   104M  /tank/test/productAbeta
# zfs promote tank/test/productAbeta
# zfs list -r tank/test
NAME                          USED  AVAIL  REFER  MOUNTPOINT
tank/test                     104M  66.2G    24K  /tank/test
tank/test/productA               0  66.2G   104M  /tank/test/productA
tank/test/productAbeta        104M  66.2G   104M  /tank/test/productAbeta
tank/test/productAbeta@today     0      -   104M  -
```

Dans la sortie `zfs list`, les informations de comptabilisation de l'espace disque du système de fichiers d'origine `productA` ont été remplacées par celles du système de fichiers `productAbeta`

Pour terminer le processus de remplacement de clone, renommez les systèmes de fichiers. Par exemple :

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
```

Vous pouvez également supprimer l'ancien système de fichiers si vous le souhaitez. Par exemple :

```
# zfs destroy tank/test/productAlegacy
```

Envoi et réception de données ZFS

La commande `zfs send` crée une représentation de flux d'un instantané qui est écrite dans la sortie standard. Un flux complet est généré par défaut. Vous pouvez rediriger la sortie vers un fichier ou un système fichier. La commande `zfs receive` crée un instantané dont le contenu est spécifié dans le flux fourni dans l'entrée standard. En cas de réception d'un flux complet, un système de fichiers est également créé. Ces commandes permettent d'envoyer les données d'instantané ZFS et de recevoir les systèmes de fichiers et les données d'instantané ZFS. Reportez - vous aux exemples dans la section suivante.

- “Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde” à la page 214
- “Envoi d'un instantané ZFS” à la page 216
- “Réception d'un instantané ZFS” à la page 217
- “Application de différentes valeurs de propriété à un flux d'instantané ZFS” à la page 219
- “Envoi et réception de flux d'instantanés ZFS complexes” à la page 221
- “Réplication distante de données ZFS” à la page 224

Les solutions de sauvegarde suivantes sont disponibles pour enregistrer les données ZFS :

- **Produits de sauvegarde d'entreprise** : si vous souhaitez disposer des fonctions suivantes, considérez une solution de sauvegarde d'entreprise :
  - Restauration fichier par fichier
  - Vérification des médias de sauvegarde
  - Gestion des médias
- **Instantanés de systèmes de fichiers et restauration d'instantanés** : exécutez les commandes `zfs snapshot` et `zfs rollback` pour créer facilement une copie d'un système de fichiers et restaurer une version précédente d'un système de fichiers, le cas échéant. Par exemple, vous pouvez utiliser cette solution pour restaurer un ou plusieurs fichiers issus d'une version précédente d'un système de fichiers.

  Pour plus d'informations sur la création et la restauration d'instantané, reportez-vous à la section “Présentation des instantanés ZFS” à la page 203.
- **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 les informations les plus récentes sur les problèmes relatifs aux produits de sauvegarde ZFS et tiers, reportez-vous aux notes de version d'Oracle Solaris 11.2.

Reconnaissance des flux d'instantané ZFS

Un instantané d'un système de fichiers ou volume ZFS est converti en flux d'instantané à l'aide de la commande `zfs send`. Ensuite, vous pouvez utiliser le flux d'instantané pour recréer un système de fichiers ou volume ZFS à l'aide de la commande `zfs receive`.

Selon les options `zfs send` utilisées pour créer le flux d'instantané, différents formats de flux sont générés.

- Flux complet : se compose du contenu intégral du jeu de données, depuis sa création jusqu'à la prise de l'instantané spécifié.

  Le flux par défaut généré par la commande `zfs send` est un flux complet. Il contient un système de fichiers ou un volume, jusqu'à et y compris l'instantané spécifié. Le flux ne contient pas d'autre instantané que celui spécifié sur la ligne de commande.
- Flux incrémentiel : se compose des différences entre deux instantanés.

Un package de flux est un type de flux contenant un ou plusieurs flux incrémentiels. Il existe trois types de packages de flux :

- Package de flux de réplication : se compose du jeu de données spécifié et de ses descendants. Il inclut tous les instantanés intermédiaires. Si l'origine d'un jeu de données cloné n'est pas un descendant de l'instantané spécifié sur la ligne de commande, le jeu de données d'origine n'est pas inclus dans le package de flux. Pour recevoir le flux, le jeu de données d'origine doit exister dans le pool de stockage de destination.

  Examinez la liste de jeux de données suivis de leur origine suivante. Nous supposerons qu'ils ont été créés dans l'ordre dans lequel ils apparaissent ci-dessous :

```
NAME                    ORIGIN
pool/a                  -
pool/a/1                -
pool/a/1@clone          -
pool/b                  -
pool/b/1                pool/a/1@clone
pool/b/1@clone2         -
pool/b/2                pool/b/1@clone2
pool/b@pre-send         -
pool/b/1@pre-send       -
pool/b/2@pre-send       -
pool/b@send             -
pool/b/1@send           -
pool/b/2@send           -
```

  Un package de flux de réplication créé en respectant la syntaxe suivante :

```
# zfs send -R pool/b@send ....
```

  Se compose des flux complets et incrémentiels suivants :

```
TYPE    SNAPSHOT                INCREMENTAL FROM
full    pool/b@pre-send         -
incr    pool/b@send             pool/b@pre-send
incr    pool/b/1@clone2         pool/a/1@clone
incr    pool/b/1@pre-send       pool/b/1@clone2
incr    pool/b/1@send           pool/b/1@send
incr    pool/b/2@pre-send       pool/b/1@clone2
incr    pool/b/2@send           pool/b/2@pre-send
```

  Dans la sortie qui précède, l'instantané `pool/a/1@clone` n'est pas inclus dans le package de flux de réplication. En l'état, ce package de flux de réplication peut uniquement être reçu dans un pool possédant déjà l'instantané `pool/a/1@clone` .

- Package de flux récursif : se compose du jeu de données spécifié et de ses descendants. A la différence des packages de flux de réplication, les instantanés intermédiaires ne sont pas inclus, sauf s'ils constituent l'origine d'un jeu de données cloné inclus dans le flux. Par défaut, si l'origine d'un jeu de données n'est pas un descendant de l'instantané spécifié

sur la ligne de commande, le comportement est le même que pour les flux de réplication. Cependant, un flux récursif autonome, comme décrit ci-après , est créé de cette façon qu'il n'existe aucune dépendance externe.

Un package de flux récursif créé en respectant la syntaxe suivante :

```
# zfs send -r pool/b@send ...
```

Se compose des flux complets et incrémentiels suivants :

```
TYPE    SNAPSHOT                INCREMENTAL FROM
full    pool/b@send             -
incr    pool/b/1@clone2         pool/a/1@clone
incr    pool/b/1@send           pool/b/1@clone2
incr    pool/b/2@send           pool/b/1@clone2
```

Dans la sortie qui précède, l'instantané `pool/a/1@clone` n'est pas inclus dans le package de flux récursif. En l'état, ce package de flux récursif peut uniquement être reçu dans un pool qui possède déjà l'instantané `pool/a/1@clone` . Ce comportement est similaire au scénario du package de flux de réplication décrit plus haut.

- Package de flux récursif autonome : ne dépend d'aucun jeu de données non inclus dans le package de flux. Le package de flux récursif créé en respectant la syntaxe suivante :

  ```
  # zfs send -rc pool/b@send ...
  ```

  Se compose des flux complets et incrémentiels suivants :

  ```
  TYPE    SNAPSHOT                INCREMENTAL FROM
  full    pool/b@send             -
  full    pool/b/1@clone2
  incr    pool/b/1@send           pool/b/1@clone2
  incr    pool/b/2@send           pool/b/1@clone2
  ```

  Notez que le flux récursif autonome possède un flux complet de l'instantané `pool/b/1@clone2`, ce qui permet la réception de l'instantané `pool/b/1` sans dépendance externe.

Envoi d'un instantané ZFS

Vous pouvez utiliser la commande `zfs send` pour envoyer une copie d'un flux d'instantané et recevoir ce flux dans un autre pool du même système ou dans un autre pool d'un système différent utilisé pour stocker les données de sauvegarde. Par exemple, pour envoyer le flux de données à un pool différent instantané du même système, utilisez la syntaxe ci-dessous :

```
# zfs send tank/dana@snap1 | zfs recv spool/ds01
```

Vous pouvez utiliser `zfs recv` en tant qu'alias pour la commande `zfs receive`.

Si vous envoyez le flux de l'instantané à un système différent, envoyez la sortie de la commande `zfs send` à la commande `ssh`. Par exemple :

```
sys1# zfs send tank/dana@snap1 | ssh sys2 zfs recv newtank/dana
```

Lors de l'envoi d'un flux complet, le système de fichiers de destination ne doit pas exister.

Vous pouvez envoyer les données incrémentielles à l'aide de l'option `zfs send- i`. Par exemple :

```
sys1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

Le premier argument (`snap1`) correspond à l'instantané le plus ancien, le second (`snap2`) à l'instantané le plus récent. Dans ce cas, le système de fichiers `newtank/dana` doit déjà exister pour que la réception incrémentielle s'effectue correctement.

Remarque - L'accès à des informations de fichier dans le système de fichiers reçu à l'origine, peut causer l'échec d'une opération de réception d'instantané incrémentale avec un message similaire à celui-ci :

```
cannot receive incremental stream of tank/dana@snap2 into newtank/dana:
most recent snapshot of tank/dana@snap2 does not match incremental source
```

Envisagez de définir la propriété `atime` sur off si vous avez besoin d'accéder à des informations de fichier dans le système de fichiers reçu à l'origine et si vous avez aussi besoin de recevoir des instantanés incrémentaux dans le système de fichiers reçu.

La source de *snap1* incrémentiel peut être spécifiée comme étant le dernier composant du nom de l'instantané. Grâce à ce raccourci, il suffit de spécifier le nom après le signe `@` pour *snap1*, qui est considéré comme provenant du même système de fichiers que *snap2*. Par exemple :

```
sys1# zfs send -i snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

Cette syntaxe de raccourci est équivalente à la syntaxe incrémentielle de l'exemple précédent.

Le message s'affiche en cas de tentative de génération d'un flux incrémentiel à partir d'un `instantané1` provenant d'un autre système de fichiers :

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Si vous devez stocker de nombreuses copies, envisagez de compresser une représentation de flux d'instantané ZFS à l'aide de la commande `gzip`. Par exemple :

```
# zfs send pool/fs@snap | gzip > backupfile.gz
```

Réception d'un instantané ZFS

Gardez les points suivants à l'esprit lorsque vous recevez un instantané du système de fichiers, procédez comme suit :

- L'instantané et le système de fichiers sont reçus.
- Le système de fichiers et tous les systèmes de fichiers descendants sont démontés.
- Les systèmes de fichiers sont inaccessibles tant qu'ils sont en cours de réception.
- Le système de fichiers d'origine à recevoir ne doit pas exister tant qu'il est en cours de transfert.
- Si ce nom existe déjà, vous pouvez utiliser la commande `zfs rename` pour renommer le système de fichiers.

Par exemple :

```
# zfs send tank/gozer@0830 > /bkups/gozer.083006
# zfs receive tank/gozer2@today < /bkups/gozer.083006
# zfs rename tank/gozer tank/gozer.old
# zfs rename tank/gozer2 tank/gozer
```

Si vous apportez des modifications au système de fichiers de destination et souhaitez effectuer un autre envoi incrémentiel d'instantané, vous devez au préalable restaurer le système de fichiers destinataire.

Voyez l'exemple suivant. Modifiez tout d'abord le système de fichiers comme suit :

```
sys2# rm newtank/dana/file.1
```

Effectuez ensuite un envoi incrémentiel de `char/dana@snap3`. Cependant, vous devez d'abord annuler (roll back) le système de fichiers destinataire pour permettre la réception du nouvel instantané incrémentiel. Vous pouvez aussi utiliser l'option `-F` pour éviter l'étape de restauration. Par exemple :

```
sys1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh sys2 zfs recv -F newtank/dana
```

Lors de la réception d'un instantané incrémentiel, le système de fichiers de destination doit déjà exister.

Si vous apportez des modifications au système de fichiers sans restaurer le système de fichiers destinataire pour permettre la réception du nouvel instantané incrémentiel, ou si vous ne spécifiez pas l'option `-F`, un message similaire au message suivant s'affiche :

```
sys1# zfs send -i tank/dana@snap4 tank/dana@snap5 | ssh sys2 zfs recv newtank/dana
cannot receive: destination has been modified since most recent snapshot
```

Les vérifications suivantes sont requises pour assurer l'exécution de l'option `-F` :

- Si l'instantané le plus récent ne correspond pas à la source incrémentielle, la restauration et la réception ne s'effectuent pas intégralement et un message d'erreur s'affiche.
- Si vous avez fourni accidentellement le nom d'un système de fichiers qui ne correspond pas à la source incrémentielle dans la commande `zfs receive`, la restauration et la réception ne s'effectuent pas correctement et le message d'erreur suivant s'affiche :

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Surveillance de la progression des flux envoyés par ZFS

La syntaxe `zfs snapshot`ci-après vous fournit un instantané ZFS avec l'estimation de la taille d'un flux de données dans le mode test. Vous pouvez utiliser l'estimation de taille pour mesurer la progresse du flux d'envoi.

Utilisez la syntaxe de test suivante pour estimer la taille du flux d'instantané sans l'envoyer.

```
# zfs snapshot -r tank/source@snap1
# zfs send -rnv tank/source@snap1
sending full stream to tank/source@snap1
sending full stream to tank/source/data@snap1
estimated stream size: 10.0G
```

Vous pouvez surveiller la progression du flux d'envoi en insérant la commande entre le contrôleur de stockage et les commandes. `pvzfs sendzfs receive`. Dans l'exemple suivant, la première commande permet d'estimer la taille du flux de données. Ensuite, le est associée à la deuxième commande est envoyé lors de l'objet d'une surveillance instantané en même temps.

```
# zfs send tank/source@snap1 | pv | zfs recv pond/target
882MB 0:00:10 [95.4MB/s] [       <=>   ]
```

Lorsque l'instantané reçu, le flux complète a été effectuée sur identifie la taille totale de surveillance de la progression reçu. Par exemple :

```
# zfs send tank/source@snap1 | pv |zfs recv pond/target
10GB 0:01:55 [88.5MG/s] [       <=>   ]
```

Application de différentes valeurs de propriété à un flux d'instantané ZFS

Vous pouvez envoyer un flux d'instantané ZFS avec une certaine valeur de propriété de système de fichiers, mais vous pouvez spécifier une valeur de propriété locale différente lorsque le flux de l'instantané est reçu. Vous pouvez également indiquer que la valeur de propriété d'origine doit être utilisée lorsque le flux d'instantané est reçu pour recréer le système de fichiers d'origine. En outre, vous pouvez désactiver une propriété de système de fichiers lorsque le flux d'instantané est reçu.

- Utilisez la commande `zfs inherit -S` pour rétablir la valeurs de propriété locale reçue, le cas échéant. Si une propriété ne reçoit aucune valeur, le comportement de la commande

`zfs inherit-S` est le même que la commande `zfs inherit` sans l'option `-S`. Si la propriété ne reçoit aucune valeur, la commande `zfs inherit` masque la valeur reçue par la valeur héritée jusqu'à ce que l'émission d'une commande `zfs inherit-S` rétablisse la valeur reçue.

- Vous pouvez utiliser la commande `zfs get -o` pour prendre en compte la nouvelle colonne `RECEIVED` ajoutée. Vous pouvez également utiliser la commande `zfs get-o all` pour ajouter toutes les colonnes, y compris la colonne `RECEIVED`.
- Vous pouvez utiliser l'option `zfs send -p` pour ajouter des propriétés dans le flux d'envoi sans l'option `-R`.
- L'option `zfs receive -e` permet d'utiliser le dernier élément du nom de l'instantané envoyé pour définir le nom du nouvel instantané. L'exemple suivant envoie l'instantané `poola/bee/cee@1` au système de fichiers `poold/eee` et utilise uniquement le dernier élément (`cee@1`) du nom de l'instantané pour créer le système de fichiers et l'instantané reçus.

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

Dans certains cas, les propriétés du système de fichiers dans un flux envoyé ne peuvent pas s'appliquer au système de fichiers récepteur ou aux propriétés du système de fichiers local, comme la valeur de propriété `mountpoint`, et risquent d'interférer avec une restauration.

Par exemple, dans le système de fichiers `tank/données`, la propriété `compression` est désactivée. Un instantané du système de fichiers `tank/data` est envoyé avec des propriétés (option `-p`) à un pool de sauvegarde et est reçu avec la propriété `compression` activée.

```
# zfs get compression tank/data
NAME       PROPERTY     VALUE     SOURCE
tank/data  compression  off       default
# zfs snapshot tank/data@snap1
# zfs send -p tank/data@snap1 | zfs recv -o compression=on -d bpool
# zfs get -o all compression bpool/data
NAME        PROPERTY     VALUE     RECEIVED  SOURCE
bpool/data  compression  on        off       local
```

Dans l'exemple, la propriété `compression` est activée lorsque l'instantané est reçu dans `bpool`. Par conséquent, pour `bpool/data`, la valeur `compression` est activée.

Si ce flux d'instantané est envoyé à un nouveau pool, `restorepool`, à des fins de récupération, vous pouvez être amené à conserver toutes les propriétés de l'instantané d'origine. Dans ce cas, vous devez utiliser la commande `zfs send -b` pour restaurer les propriétés de l'instantané d'origine. Par exemple :

```
# zfs send -b bpool/data@snap1 | zfs recv -d restorepool
# zfs get -o all compression restorepool/data
NAME              PROPERTY     VALUE     RECEIVED  SOURCE
restorepool/data  compression  off       off       received
```

Dans l'exemple, la valeur de compression est `off`, elle représente la valeur de compression de l'instantané du système de fichiers `tank/data` d'origine.

Si vous disposez d'une valeur de propriété de système de fichiers local dans un flux d'instantané et que vous souhaitez désactiver la propriété lors de sa réception, utilisez la commande `zfs receive -x`. Par exemple, la commande suivante envoie un flux d'instantané récursif des systèmes de fichiers du répertoire d'accueil avec toutes les propriétés de système de fichiers réservées à un pool de sauvegarde, mais sans les valeurs de propriété du quota.

```
# zfs send -R tank/home@snap1 | zfs recv -x quota bpool/home
# zfs get -r quota bpool/home
NAME                   PROPERTY  VALUE  SOURCE
bpool/home             quota     none   local
bpool/home@snap1       quota     -      -
bpool/home/lori        quota     none   default
bpool/home/lori@snap1  quota     -      -
bpool/home/mark        quota     none   default
bpool/home/mark@snap1  quota     -      -
```

Si l'instantané récursif n'a pas été reçu avec l'option `-x`, la propriété de quota doit être définie dans les systèmes de fichiers reçus.

```
# zfs send -R tank/home@snap1 | zfs recv bpool/home
# zfs get -r quota bpool/home
NAME                   PROPERTY  VALUE  SOURCE
bpool/home             quota     none   received
bpool/home@snap1       quota     -      -
bpool/home/lori        quota     10G    received
bpool/home/lori@snap1  quota     -      -
bpool/home/mark        quota     10G    received
bpool/home/mark@snap1  quota     -      -
```

Envoi et réception de flux d'instantanés ZFS complexes

Cette section décrit l'utilisation des options `zfs send-I` et `-R` pour envoyer et recevoir des flux d'instantanés plus complexes.

Gardez les points suivants à l'esprit lors de l'envoi et de la réception de flux d'instantanés ZFS complexes :

- Utilisez l'option `zfs send -I` pour envoyer tous les flux incrémentiels d'un instantané à un instantané cumulé. Vous pouvez également utiliser cette option pour envoyer un flux incrémentiel de l'instantané d'origine pour créer un clone. L'instantané d'origine doit déjà exister sur le côté récepteur afin d'accepter le flux incrémentiel.
- Utilisez l'option `zfs send -R` pour envoyer un flux de réplication de tous les systèmes de fichiers descendants. Une fois le flux de réplication reçu, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.
- Lorsque l'option `zfs send -r` est utilisée sans l'option `-c` et lorsque l'option `zfs send -R` est utilisée, les packages de flux omettent dans certains cas l'`origin` des clones. Pour plus d'informations, reportez-vous à la section “Reconnaissance des flux d'instantané ZFS” à la page 214.
- Vous pouvez utiliser les deux options pour envoyer un flux de réplication incrémentiel.
  - Les modifications des propriétés sont conservées, tout comme les opérations `rename` et `destroy` des instantanés et des systèmes de fichiers.
  - Si l'option `zfs recv -F` n'est pas spécifiée lors de la réception du flux de réplication, les opérations `destroy` du jeu de données sont ignorées. La syntaxe de `zfs recv -F` dans ce cas peut conserve également sa signification de *récupération le cas échéant*.
  - Tout comme dans les autres cas `- i` ou `-I` (autres que `zfs send-R`), si l'option `-I` est utilisée, tous les instantanés créés entre `snapA` et `snapD` sont envoyés. Si l'option `-i` est utilisée, seul `snapD` (pour tous les descendants) est envoyé.
- Pour recevoir ces nouveaux types de flux `zfs send`, le système récepteur doit exécuter une version du logiciel capable de les envoyer. La version des flux est incrémentée.

  Vous pouvez cependant accéder à des flux d'anciennes versions de pool en utilisant une version plus récente du logiciel. Vous pouvez par exemple envoyer et recevoir des flux créés à l'aide des nouvelles options à partir d'un pool de la version 3. Vous devez par contre exécuter un logiciel récent pour recevoir un flux envoyé avec les nouvelles options.

EXEMPLE 6-1 Envoi et réception de flux d'instantanés ZFS complexes

Plusieurs instantanés incrémentiels peuvent être regroupés en un seul instantané à l'aide de l'option `zfs send -I`. Par exemple :

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

Vous pouvez ensuite supprimer `snapB`, `snapC` et `snapD`.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Pour recevoir les instantanés combinés, vous devez utiliser la commande suivante :

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

Vous pouvez également utiliser la commande `zfs send -I` pour regrouper un instantané et un clone d'instantané en un nouveau jeu de données. Par exemple :

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

Vous pouvez utiliser la commande `zfs send -R` pour répliquer un système de fichiers ZFS et tous ses systèmes de fichiers descendants, jusqu'à l'instantané nommé. Une fois ce flux reçu, les propriétés, instantanés, systèmes de fichiers descendants et clones sont conservés.

Dans l'exemple suivant, des instantanés des systèmes de fichiers utilisateur sont créés. Un flux de réplication de tous les instantanés utilisateur est créé. Les systèmes de fichiers et instantanés d'origine sont ensuite détruits et récupérés.

```
# zfs snapshot -r users@today
# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
users                187K  33.2G    22K  /users
users@today             0      -    22K  -
users/user1           18K  33.2G    18K  /users/user1
users/user1@today       0      -    18K  -
users/user2           18K  33.2G    18K  /users/user2
users/user2@today       0      -    18K  -
users/user3           18K  33.2G    18K  /users/user3
users/user3@today       0      -    18K  -
# zfs send -R users@today > /snaps/users-R
# zfs destroy -r users
# zfs receive -F -d users < /snaps/users-R
# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
users                196K  33.2G    22K  /users
users@today             0      -    22K  -
users/user1           18K  33.2G    18K  /users/user1
users/user1@today       0      -    18K  -
users/user2           18K  33.2G    18K  /users/user2
users/user2@today       0      -    18K  -
users/user3           18K  33.2G    18K  /users/user3
```

```
users/user3@today      0      -    18K  -
```

Dans l'exemple suivant, la commande `zfs send -R` a été utilisée pour répliquer le système de fichiers `users` et ses descendants et pour envoyer le flux répliqué vers un autre pool, `users2`.

```
# zfs create users2 mirror c0t1d0 c1t1d0
# zfs receive -F -d users2 < /snaps/users-R
# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
users                224K  33.2G    22K  /users
users@today             0      -    22K  -
users/user1           33K  33.2G    18K  /users/user1
users/user1@today     15K      -    18K  -
users/user2           18K  33.2G    18K  /users/user2
users/user2@today       0      -    18K  -
users/user3           18K  33.2G    18K  /users/user3
users/user3@today       0      -    18K  -
users2               188K  16.5G    22K  /users2
users2@today            0      -    22K  -
users2/user1          18K  16.5G    18K  /users2/user1
users2/user1@today      0      -    18K  -
users2/user2          18K  16.5G    18K  /users2/user2
users2/user2@today      0      -    18K  -
users2/user3          18K  16.5G    18K  /users2/user3
users2/user3@today      0      -    18K  -
```

Réplication distante de données ZFS

Les commandes `zfs send` et `zfs recv` permettent d'effectuer une copie distante d'une représentation de flux d'instantané d'un système vers un autre. Par exemple :

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Cette commande envoie les données de l'instantané `tank/cindy@today` et les reçoit dans le système de fichiers `sandbox/restfs`. La commande suivante crée également un instantané `restfs@aujourd'hui` sur le système `newsys`. Dans cet exemple, l'utilisateur a été configuré pour utiliser `ssh` dans le système distant.

♦ ♦ ♦ CHAPITRE 7

Utilisation des ACL et des attributs pour protéger les fichiers Oracle Solaris ZFS

Ce chapitre décrit l'utilisation des listes de contrôle d'accès (ACL, Access Control List) pour protéger les fichiers ZFS en accordant des autorisations à un niveau de granularité plus fin que les autorisations UNIX standard.

Ce chapitre contient les sections suivantes :

Modèle ACL Solaris

Les versions précédentes de Solaris assuraient la prise en charge d'une implémentation ACL reposant principalement sur la spécification POSIX-draft ACL. Les ACL basées sur POSIX-draft sont utilisées pour protéger les fichiers UFS et sont traduites par les versions de NFS antérieures à NFSv4.

Grâce à l'introduction de NFSv4, un nouveau modèle d'ACL assure entièrement la prise en charge de l'interopérabilité qu'offre NFSv4 entre les clients UNIX et non UNIX. La nouvelle implémentation d'ACL, telle que définie dans les spécifications NFSv4, fournit des sémantiques bien plus riches, basées sur des ACL NT.

Les différences principales du nouveau modèle d'ACL sont les suivantes :

- Modèle basé sur la spécification NFSv4 et similaire aux ACL de type NT.
- Jeu de privilèges d'accès bien plus granulaire. Pour plus d'informations, reportez-vous au Tableau 7-2, "Privilèges d'accès d'ACL".

- Configuration et affichage avec les commandes `chmod` et `ls`, et non les commandes `setfacl` et `getfacl`.
- Sémantique d'héritage bien plus riche pour déterminer comment les privilèges d'accès sont appliqués d'un répertoire à un sous-répertoire, et ainsi de suite. Pour plus d'informations, reportez-vous à la section “Héritage ACL” à la page 230.

Les deux modèles d'ACL assurent un contrôle d'accès à un niveau de granularité plus fin que celui disponible avec les autorisations de fichier standard. De façon similaire aux listes de contrôle d'accès POSIX-draft, les nouvelles ACL se composent de plusieurs ACE (Access Control Entry, entrées de contrôle d'accès).

Les ACL de type POSIX-draft utilisent une seule entrée pour définir quelles autorisations sont accordées et lesquelles sont refusées. Le nouveau modèle d'ACL dispose de deux types d'ACE qui affectent la vérification d'accès : `ALLOW` et `DENY`. Il est en soi impossible de déduire ACE à partir de n'importe quel unique qui définit un ensemble d'autorisations si les autorisations qui n'ont pas été définies dans cette ACE sont ou non accordées.

La conversion entre les ACL NFSv4 et les ACL POSIX-draft s'effectue comme suit :

- Si vous employez un utilitaire compatible avec les ACL, les commandes `cp`, `mv`, `tar`, `cpio` ou `rcp`, (par exemple) pour transférer des fichiers UFS avec des ACL vers un système de fichiers ZFS, les ACL POSIX-draft sont converties en ACL NFSv4 équivalentes.
- Les ACL NFSv4 sont converties en ACL POSIX-draft. Un message tel que le suivant s'affiche si une ACL NFSv4 n'est pas convertie en ACL POSIX-draft :

  ```
  # cp -p filea /var/tmp
  cp: failed to set acl entries on /var/tmp/filea
  ```

- Si vous créez une archive `cpio` ou `tar` UFS avec l'option de conservation des ACL (`tar-p` ou `cpio-P`) dans un système exécutant la version actuelle de Solaris, les ACL sont perdues en cas d'extraction de l'archive sur un système exécutant une version précédente de Solaris.

  Tous les fichiers sont extraits avec les modes de fichier corrects, mais les entrées d'ACL sont ignorées.
- Vous pouvez utiliser la commande `ufsrestore` pour restaurer des données dans un système de fichiers ZFS. Si les données d'origine incluent des ACL POSIX-style, elles sont converties en ACL NFSv4-style.
- En cas de tentative de configuration d'une ACL SFSv4 dans un fichier UFS, un message tel que le suivant s'affiche :

  ```
  chmod: ERROR: ACL type's are different
  ```

- En cas de tentative de configuration d'une ACL POSIX dans un fichier ZFS, un message tel que le suivant s'affiche :

  ```
  # getfacl filea
  File system doesn't support aclent_t style ACL's.
  See acl(5) for more information on Solaris ACL support.
  ```

Pour obtenir des informations sur les autres limitations des ACL et des produits de sauvegarde, reportez-vous à la section “Enregistrement de données ZFS à l'aide d'autres produits de sauvegarde” à la page 214.

Descriptions de syntaxe pour la configuration des ACL

Deux formats ACL de base sont fournis comme suit :

- **ACL triviales** : Contient uniquement des entrées `user`, `group` et `owner` UNIX traditionnelles.

  Utilisez la syntaxe de commande suivante pour définir d'ACL triviales.

  ```
  chmod [options] A[index]{+|=}owner@ |group@ |everyone@: \
    access-permissions/...[:inheritance-flags]:deny | allow file

  chmod [options] A-owner@, group@, everyone@: \
    access-permissions/...[:inheritance-flags]:deny | allow file ...

  chmod [options] A[index]- file
  ```

- **ACL non triviale** – Contient plus d'entrées qu'owner, group et everyone, inclut des indicateurs d'héritage ou les entrées sont triées d'une manière non traditionnelle.

  Utilisez la syntaxe de commande suivante pour définir d'ACL non triviales.

  ```
  chmod [options] A[index]{+|=}user|group:name: \
  access-permissions/...[:inheritance-flags]:deny | allow file

  chmod [options] A-user|group:name: \
  access-permissions/...[:inheritance-flags]:deny | allow file ...

  chmod [options] A[index]- file
  ```

La liste suivante explique les options qui sont utilisés dans les commandes permettant de définir la configuration de contrôle d'accès (ACL) et non triviales.

owner@, group@, everyone@	Identifie le *type d'entrée d'ACL* pour la syntaxe d'ACL triviale. Pour obtenir une description des *types d'entrées d'ACL*, reportez-vous au Tableau 7-1, “Types d'entrées d'ACL”.
utilisateur ou groupe :*ID-entrée-ACL=nomutilisateur ou nomgroupe*	Identifie le *type d'entrée d'ACL* pour la syntaxe d'ACL explicite. Le *type d'entrée d'ACL* pour l'utilisateur et le groupe doit également contenir l'*ID d'entrée d'ACL*, le *nom d'utilisateur* ou le *nom de groupe*. Pour obtenir une description des *types d'entrées d'ACL*, reportez-vous au Tableau 7-1, “Types d'entrées d'ACL”.

autorisations-d'accès/.../ — Identifie les autorisations d'accès accordées ou refusées. Pour obtenir une description des privilèges d'accès d'ACL, reportez-vous au Tableau 7-2, “Privilèges d'accès d'ACL”.

indicateurs-héritage — Identifie une liste optionnelle d'indicateurs d'héritage d'ACL. Pour une description des indicateurs d'héritage d'ACL, reportez-vous au Tableau 7-4, “Indicateurs d'héritage d'ACL”.

`deny` | `allow` — Détermine si les autorisations d'accès sont accordées ou refusées.

Dans l'exemple suivant, il n'existe aucune valeur `ACL-entry-ID` pour `owner@`, `group@` ou `everyone@`.

```
group@:write_data/append_data/execute:deny
```

L'exemple suivant inclut un `ID d'entrée d'ACL` car un utilisateur spécifique (`type d'entrée d'ACL`) est inclus dans la liste.

```
0:user:joe:list_directory/read_data/execute:allow
```

Lorsqu'une entrée d'ACL s'affiche, elle est similaire à celle-ci :

```
2:group@:write_data/append_data/execute:deny
```

La désignation `2` ou `ID d'index` dans cet exemple identifie l'entrée d'ACL dans la plus grande ACL, qui peut présenter plusieurs entrées pour le propriétaire, des UID spécifiques, un groupe et pour tous. Vous pouvez spécifier l'*ID d'index* avec la commande `chmod` pour identifier la partie de l'ACL que vous souhaitez modifier. Par exemple, vous pouvez identifier l'ID d'index 3 par `A3` dans la commande `chmod` comme ci-dessous :

```
chmod A3=user:venkman:read_acl:allow filename
```

Le tableau suivant décrit les types de format ACL des représentations, autrement dit la contrevaleur du propriétaire, du groupe et autres.

TABLEAU 7-1 Types d'entrées d'ACL

Type d'entrée d'ACL	Description
`owner@`	Spécifie l'accès accordé au propriétaire de l'objet.
`group@`	Spécifie l'accès accordé au groupe propriétaire de l'objet.
`everyone@`	Spécifie l'accès accordé à tout utilisateur ou groupe ne correspondant à aucune autre entrée d'ACL.
`utilisateur`	Avec un nom d'utilisateur, spécifie l'accès accordé à un utilisateur supplémentaire de l'objet. Doit inclure l'*ID d'entrée d'ACL* qui contient un *username* ou un *userID*. Le type d'entrée d'ACL est incorrect si la valeur n'est ni un UID numérique, ni un *username*.
`groupe`	Avec un nom de groupe, spécifie l'accès accordé à un utilisateur supplémentaire de l'objet. Doit inclure l'*ID d'entrée d'ACL* qui contient un *groupname* ou un *groupID*. Le type d'entrée d'ACL est incorrect si la valeur n'est ni un GID numérique, ni un *groupname*.

Les privilèges d'accès sont décrits dans le tableau suivant.

TABLEAU 7-2 Privilèges d'accès d'ACL

Privilège d'accès	Privilège d'accès compact	Description
add_file	w	Autorisation d'ajouter un fichier à un répertoire.
add_subdirectory	p	Dans un répertoire, autorisation de créer un sous-répertoire.
append_data	p	Non implémentée actuellement.
delete	d	Autorisation de supprimer un fichier. Pour plus d'informations sur le comportement spécifique de l'autorisation delete_child , reportez-vous au Tableau 7-3, “Comportement d'autorisation ACL delete et delete_child”.
delete_child	D	Autorisation de supprimer un fichier ou un répertoire au sein d'un répertoire. Pour plus d'informations sur le comportement spécifique de l'autorisation delete_child , reportez-vous au Tableau 7-3, “Comportement d'autorisation ACL delete et delete_child”.
execute	x	Autorisation d'exécuter un fichier ou d'effectuer une recherche dans le contenu d'un répertoire.
list_directory	r	Autorisation de dresser la liste du contenu d'un répertoire.
read_acl	c	Autorisation de lire l'ACL (ls).
read_attributes	a	Autorisation de lire les attributs de base (non ACL) d'un fichier. Considérez les attributs de base comme les attributs de niveau stat. L'autorisation de ce bit de masque d'accès signifie que l'entité peut exécuter ls(1) et stat(2).
read_data	r	Autorisation de lire le contenu du fichier.
read_xattr	R	Autorisation de lire les attributs étendus d'un fichier ou d'effectuer une recherche dans le répertoire d'attributs étendus d'un fichier.
synchronize	s	Non implémentée actuellement.
write_xattr	W	Autorisation de créer des attributs étendus ou d'écrire dans le répertoire d'attributs étendus. L'attribution de cette autorisation à un utilisateur signifie que ce dernier peut créer un répertoire d'attributs étendus pour un fichier. Les autorisations du fichier d'attributs contrôlent l'accès de l'utilisateur à l'attribut.
write_data	w	Autorisation de modifier ou de remplacer le contenu d'un fichier.
write_attributes	R	Autorisation de remplacer les durées associées à un fichier ou un répertoire par une valeur arbitraire.
write_acl	C	Autorisation d'écriture sur l'ACL ou capacité de la modifier à l'aide de la commande chmod.
write_owner	O	Autorisation de modifier le propriétaire ou le groupe d'un fichier. Ou capacité d'exécuter les commandes chown ou chgrp sur le fichier. Autorisation de devenir propriétaire d'un fichier ou autorisation de définir la propriété de groupe du fichier sur un groupe dont fait partie l'utilisateur. Le privilège PRIV_FILE_CHOWN est requis pour définir la propriété de fichier ou de groupe sur un groupe ou un utilisateur arbitraire.

Le tableau suivant fournit des détails supplémentaires sur les comportements `delete` et `delete_child` d'ACL.

TABLEAU 7-3 Comportement d'autorisation ACL `delete` et `delete_child`

Droits d'accès au répertoire parent	Autorisations d'objet cible		
	L'ACL autorise la suppression	L'ACL refuse la suppression	Autorisation de suppression non spécifiée
L'ACL autorise `delete_child`	Autorisation	Autorisation	Autorisation
L'ACL refuse `delete_child`	Autorisation	Rejeter	Rejeter
L'ACL autorise uniquement `write` et `execute`	Autorisation	Autorisation	Autorisation
L'ACL refuse `write` et `execute`	Autorisation	Rejeter	Rejeter

Jeux d'ACL ZFS

Au lieu de définir séparément des autorisations individuelles, il est possible d'appliquer les combinaisons d'ACL suivantes par le biais d'un *jeu d'ACL*. Les jeux d'ACL suivants sont disponibles :

Nom de jeu d'ACL	Autorisations d'ACL incluses
`full_set`	Toutes les autorisations
`modify_set`	Toutes les autorisations à l'exception de `write_acl` et `write_owner`
`read_set`	`read_data`, `read_attributes`, `read_xattr` et `read_acl`
`write_set`	`write_data`, `append_data`, `write_attributes` et `write_xattr`

Ces jeux d'ACL sont prédéfinis et ne peuvent pas être modifiés

Héritage ACL

Le but de l'héritage ACL est qu'un fichier ou répertoire récemment créé hérite des ACL qui leur sont destinées, tout en tenant compte des bits d'autorisation existants dans le répertoire parent.

Par défaut, les ACL ne sont pas propagées. Si vous configurez une ACL non triviale dans un répertoire, aucun répertoire enfant n'en hérite. Vous devez spécifier l'héritage d'une ACL dans un fichier ou un répertoire.

Les indicateurs d'héritage facultatifs sont décrits dans le tableau ci-dessous.

Remarque - Actuellement, les indicateurs `successful_access`, `failed_access` et `inherited` s'appliquent uniquement au serveur SMB.

TABLEAU 7-4 Indicateurs d'héritage d'ACL

Indicateur d'héritage	Indicateur d'héritage compact	Description
`file_inherit`	`f`	Hérite de l'ACL uniquement à partir du répertoire parent vers les fichiers du répertoire.
`dir_inherit`	`d`	Hérite de l'ACL uniquement à partir du répertoire parent vers les sous-répertoires du répertoire.
`inherit_only`	`i`	Hérite de l'ACL à partir du répertoire parent mais ne s'applique qu'aux fichiers et sous-répertoires récemment créés, pas au répertoire lui-même. Cet indicateur requiert les indicateurs `file_inherit` et/ou `dir_inherit` afin de spécifier ce qui doit être hérité.
`no_propagate`	`t`	N'hérite que de l'ACL provenant du répertoire parent vers le contenu de premier niveau du répertoire, et non les contenus de second niveau et suivants. Cet indicateur requiert les indicateurs `file_inherit` et/ou `dir_inherit` afin de spécifier ce qui doit être hérité.
`-`	S/O	Aucune autorisation n'est accordée.
`successful_access`	`S`	Indique si une alarme ou un enregistrement d'audit doit être initié lorsqu'un accès réussit. Cet indicateur est utilisé avec les types d'ACE (entrées de contrôle d'accès) d'audit ou d'alarme.
`failed_access`	`F`	Indique si une alarme ou un enregistrement d'audit doit être lancé lorsqu'un accès échoue. Cet indicateur est utilisé avec les types d'ACE (entrées de contrôle d'accès) d'audit ou d'alarme.
`inherited`	`I`	Indique qu'une ACE a été héritée.

De plus, vous pouvez configurer une stratégie d'héritage d'ACL par défaut plus ou moins stricte sur le système de fichiers à l'aide de la propriété de système de fichiers `aclinherit`. Pour plus d'informations, consultez la section suivante.

Propriétés ACL

Le système de fichiers ZFS inclut les propriétés d'ACL suivantes permettant de déterminer le comportement spécifique de l'héritage d'ACL et des interactions d'ACL avec les opérations `chmod`.

- `aclinherit` : détermine le comportement d'héritage d'ACL. Les valeurs possibles sont les suivantes :
  - `discard` : pour les nouveaux objets, aucune entrée d'ACL n'est héritée lors de la création d'un fichier ou d'un répertoire. L'ACL dans le fichier ou le répertoire est égale au mode d'autorisation du fichier ou répertoire.

- `noallow` : pour les nouveaux objets, seules les entrées d'ACL héritables dont le type d'accès est `deny` sont héritées.
- `restricted` : pour les nouveaux objets, les autorisations `write_owner` et `write_acl` sont supprimées lorsqu'une entrée d'ACL est héritée.
- `passthrough` : lorsqu'une valeur de propriété est définie sur `passthrough`, les fichiers sont créés dans un mode déterminé par les ACE héritées. Si aucune ACE pouvant être héritée n'affecte le mode, ce mode est alors défini en fonction du mode demandé à partir de l'application.
- `passthrough-x` : a la même sémantique que `passthrough`, si ce n'est que lorsque `passthrough-x` est activé, les fichiers sont créés avec l'autorisation d'exécution (`x`), mais uniquement si l'autorisation d'exécution est définie en mode de création de fichier et dans une entrée de contrôle d'accès (ACE) pouvant être héritée et qui affecte le mode.

Le mode par défaut de `aclinherit` est `restricted`.

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

  Le mode par défaut pour `aclmode` est `discard`.

  Pour plus d'informations sur l'utilisation de la propriété `aclmode`, reportez-vous à l'Exemple 7-14, “Interactions entre les ACL et les opérations `chmod` sur les fichiers ZFS”.

Configuration d'ACL dans des fichiers ZFS

Dans la mesure où elles sont implémentées avec ZFS, les ACL se composent d'un tableau d'entrées d'ACL. ZFS fournit un modèle d'ACL *pur*, dans lequel tous les fichiers présentent

une ACL. En règle générale, cette liste est *triviale* dans la mesure où elle ne représente que les entrées `propriétaire/groupe/autre` UNIX classiques.

Les fichiers ZFS disposent toujours de bits d'autorisation et d'un mode, mais ces valeurs constituent plutôt un cache de ce que représente une ACL. Par conséquent, si vous modifiez les autorisations du fichier, son ACL est mise à jour en conséquence. En outre, si vous supprimez une ACL non triviale qui accordait à un utilisateur l'accès à un fichier ou à un répertoire, il est possible que cet utilisateur y ait toujours accès en raison des bits d'autorisation qui accordent l'accès à un groupe ou à tous les utilisateurs. L'ensemble des décisions de contrôle d'accès est régi par les autorisations représentées dans l'ACL d'un fichier ou d'un répertoire.

Les règles principales d'accès aux ACL dans un fichier ZFS sont comme suit :

- ZFS traite les entrées d'ACL dans l'ordre dans lesquelles elles sont répertoriées dans l'ACL, en partant du haut.
- Seules les entrées d'ACL disposant d'un " who " correspondant au demandeur d'accès sont traitées.
- Une fois l'autorisation accordée, cette dernière ne peut plus être refusée par la suite par une entrée d'ACL dans le même jeu de d'autorisations d'ACL.
- Le propriétaire du fichier dispose de l'autorisation `write_acl` de façon inconditionnelle, même si celle-ci est explicitement refusée. Dans le cas contraire, toute autorisation non spécifiée est refusée.

  Dans les cas d'autorisations deny ou lorsqu'une autorisation d'accès est manquante, le sous-système de privilèges détermine la demande d'accès accordée pour le propriétaire du fichier ou pour le superutilisateur. Ce mécanisme évite que les propriétaires de fichiers puissent accéder à leurs fichiers et permet aux superutilisateurs de modifier les fichiers à des fins de récupération.

Si vous configurez une ACL non triviale dans un répertoire, les enfants du répertoire n'en héritent pas automatiquement. Si vous configurez une ACL non triviale, et souhaitez qu'elle soit héritée par les enfants du répertoire, vous devez utiliser les indicateurs d'héritage d'ACL. Pour plus d'informations, reportez-vous au Tableau 7-4, “Indicateurs d'héritage d'ACL” et à la section “Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé” à la page 240.

Lorsque vous créez un fichier, en fonction de la valeur `umask`, une ACL triviale par défaut, similaire à la suivante, est appliquée :

```
$ ls -v file.1
-rw-r--r--   1 root     root      206663 Jun 23 15:06 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Chaque fichier catégorie d'utilisateur (`owner@`, `group@`, `everyone@`) dispose d'une entrée ACL dans l'exemple ci-dessous.

Voici une description de l'ACL de ce fichier :

`0:owner@` Le propriétaire peut lire et modifier le contenu du fichier (`read_data/write_data/append_data`). Il peut également modifier les attributs du fichier tels que les horodatages, les attributs étendus et les ACL (`write_xattr/write_attributes /write_acl`). De plus, le propriétaire peut modifier la propriété du fichier (`write_owner`).

L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

`1:group@` Les autorisations de lecture du fichier et de ses attributs sont attribuées au groupe (`read_data/read_xattr/read_attributes/read_acl:allow`).

`2:everyone@` Les autorisations de lecture du fichier et de ses attributs sont attribués à toute personne ne correspondant ni à un utilisateur ni à un groupe (`read_data/read_xattr/read_attributes/read_acl/synchronize:allow`). L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

Lorsqu'un répertoire est créé, en fonction de la valeur `umask`, l'ACL par défaut du répertoire est similaire à l'exemple suivant :

```
$ ls -dv dir.1
drwxr-xr-x   2 root     root           2 Jul 20 13:44 dir.1
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Une description de ce répertoire est le suivant :

`0:owner@` Le propriétaire peut lire et modifier le contenu du répertoire (`list_directory/read_data/add_file/write_data/add_subdirectory /append_data`) et lire et modifier les attributs du fichier tels que les horodatages, les attributs étendus et les ACL (`/read_xattr/write_xattr/read_attributes/write_attributes/read_acl/write_acl`). En outre, le propriétaire peut faire des recherches dans le contenu (`execute`), supprimer un fichier ou un répertoire (`delete_child`) et modifier la possession du répertoire (`write_owner:allow`).

L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

`1:group@` Le groupe peut répertorier et lire le contenu et les attributs du répertoire. De plus, le groupe dispose d'autorisations d'exécution pour effectuer des

recherches dans le contenu du répertoire (`list_directory/read_data/read_xattr/execute/read_attributes /read_acl/synchronize:allow`).

`2:everyone@` Toute personne n'étant ni un utilisateur ni un groupe dispose d'autorisations de lecture et d'exécution sur le contenu et les attributs du répertoire (`list_directory/read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow`). L'autorisation d'accès `synchronize` n'est actuellement pas implémentée.

Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé

Vous pouvez modifier les ACL dans des fichiers ZFS à l'aide de la commande `chmod`. La syntaxe `chmod` suivante pour la modification de l'ACL utilise la *spécification acl* pour identifier le format de la liste. Pour obtenir une description de *acl-specification*, reportez-vous à "Descriptions de syntaxe pour la configuration des ACL " à la page 227.

- Ajout d'entrées d'ACL
  - Ajout d'une entrée d'ACL pour un utilisateur

    ```
    % chmod A+acl-specification  filename
    ```
  - Ajout d'une entrée d'ACL par *index-ID*

    ```
    % chmod Aindex-ID+acl-specification  filename
    ```

    Cette syntaxe insère la nouvelle entrée d'ACL à l'emplacement d'*index-ID* spécifié.
- Remplacement d'une entrée d'ACL

  ```
  % chmod A=acl-specification  filename
  ```

  ```
  % chmod Aindex-ID=acl-specification  filename
  ```
- Suppression d'entrées d'ACL
  - Suppression d'une entrée d'ACL par l'*index-ID*

    ```
    % chmod Aindex-ID-  filename
    ```
  - Suppression d'une entrée d'ACL par utilisateur

    ```
    % chmod A-acl-specification  filename
    ```
  - Suppression de la totalité des ACE non triviales d'un fichier

    ```
    % chmod A-  filename
    ```

Les informations détaillées de l'ACL s'affichent à l'aide de la commande `ls- v`. Par exemple :

```
# ls -v file.1
-rw-r--r--   1 root     root      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Pour obtenir des informations sur l'utilisation du format d'ACL compact, consultez “Configuration et affichage d'ACL dans des fichiers ZFS en format compact” à la page 246.

EXEMPLE 7-1 Modification des ACL triviales dans des fichiers ZFS

Cette section fournit des exemples de configuration et d'affichage d'ACL triviale, ce qui signifie que seules les entrées UNIX traditionnelles, user, group et other sont incluses dans l'ACL.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.1` :

```
# ls -v file.1
-rw-r--r--   1 root     root      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Dans l'exemple suivant, les autorisations `write_data` sont accordées au groupe `group@`.

```
# chmod A1=group@:read_data/write_data:allow file.1
# ls -v file.1
-rw-rw-r--   1 root     root      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/write_data:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Dans l'exemple suivant, les autorisations du fichier `file.1` sont reconfigurées sur 644.

```
# chmod 644 file.1
# ls -v file.1
-rw-r--r--   1 root     root      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
```

```
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

EXEMPLE 7-2 Configuration d'ACL non triviales dans des fichiers ZFS

Cette section fournit des exemples de configuration et d'affichage d'ACL non triviales.

Dans l'exemple suivant, les autorisations `read_data/execute` sont ajoutées à l'utilisateur `gozer` dans le répertoire `test.dir`.

```
# chmod A+user:joe:read_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:23 test.dir
0:user:joe:list_directory/read_data/execute:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Dans l'exemple suivant, les autorisations `read_data/execute` sont retirées à l'utilisateur `joe`.

```
# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x   2 root     root           2 Jul 20 14:23 test.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

EXEMPLE 7-3 Interactions entre les ACL et les autorisations dans les fichiers ZFS

Les exemples d'ACL suivants illustrent les interactions entre la configuration des ACL et la modification successive des bits d'autorisation du répertoire ou du fichier.

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.2`:

```
# ls -v file.2
-rw-r--r--   1 root     root        2693 Jul 20 14:26 file.2
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
```

```
:allow
```

Dans l'exemple suivant, les autorisations d'ACL `allow` sont supprimées de `everyone@`.

```
# chmod A2- file.2
# ls -v file.2
-rw-r-----   1 root     root        2693 Jul 20 14:26 file.2
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
```

Dans cette sortie, les bits d'autorisation du fichier sont réinitialisés de 644 à 640. Les autorisations de lecture de `everyone@` ont été supprimées des bits d'autorisation du fichier lorsque les autorisations "allow" des ACL ont été supprimées de `everyone@`.

Dans l'exemple suivant, l'ACL existante est remplacée par des autorisations `read_data/write_data` pour `everyone@`.

```
# chmod A=everyone@:read_data/write_data:allow file.3
# ls -v file.3
-rw-rw-rw-   1 root     root        2440 Jul 20 14:28 file.3
0:everyone@:read_data/write_data:allow
```

Dans cette sortie, la syntaxe `chmod` remplace effectivement l'ACL existante par les autorisations `read_data/write_data:allow` pour les autorisations de lecture/écriture pour le propriétaire, le groupe et `everyone@`. Dans ce modèle, `everyone@` spécifie l'accès à tout utilisateur ou groupe. Dans la mesure où aucune entrée d'ACL `owner@` ou `group@` n'existe pour ignorer les autorisations pour l'utilisateur ou le groupe, les bits d'autorisation sont définis sur 666.

Dans l'exemple suivant, l'ACL existante est remplacée par des autorisations de lecture pour l'utilisateur `joe`.

```
# chmod A=user:joe:read_data:allow file.3
# ls -v file.3
----------+  1 root     root        2440 Jul 20 14:28 file.3
0:user:joe:read_data:allow
```

Dans cette sortie, les autorisations de fichier sont calculées pour être 000 car aucune entrée d'ACL n'existe pour `owner@`, `group@`, ou `everyone@`, qui représentent les composant d'autorisation classiques d'un fichier. Le propriétaire du fichier peut résoudre ce problème en réinitialisant les autorisations (et l'ACL) comme suit :

```
# chmod 655 file.3
# ls -v file.3
-rw-r-xr-x   1 root     root        2440 Jul 20 14:28 file.3
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/execute/read_attributes/read_acl
/synchronize:allow
```

```
3:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
/synchronize:allow
```

EXEMPLE 7-4 Restauration des ACL triviales dans des fichiers ZFS

Vous pouvez utiliser la commande `chmod` pour supprimer toutes les ACL non triviales dans un fichier ou un répertoire.

Dans l'exemple suivant, deux ACE non triviales existent dans `test5.dir`.

```
# ls -dv test5.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:32 test5.dir
0:user:lp:read_data:file_inherit:deny
1:user:joe:read_data:file_inherit:deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
3:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
4:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Dans l'exemple suivant, les ACL non triviales pour les utilisateurs `joe` et `lp` sont supprimées. L'ACL restante contient les valeurs par défaut de `owner@`, `group@` et `everyone@`.

```
# chmod A- test5.dir
# ls -dv test5.dir
drwxr-xr-x   2 root     root           2 Jul 20 14:32 test5.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

EXEMPLE 7-5 Application d'un jeu d'ACL à des fichiers ZFS

Des jeux d'ACL sont fournis pour vous éviter d'avoir à appliquer séparément les autorisations d'ACL. Pour une description des jeux d'ACL, reportez-vous à la section “Jeux d'ACL ZFS” à la page 230.

Vous pouvez par exemple appliquer le jeu `read_set` comme suit :

```
# chmod A+user:bob:read_set:allow file.1
# ls -v file.1
-r--r--r--+  1 root     root      206695 Jul 20 13:43 file.1
0:user:bob:read_data/read_xattr/read_attributes/read_acl:allow
```

```
1:owner@:read_data/read_xattr/write_xattr/read_attributes
/write_attributes/read_acl/write_acl/write_owner/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Vous pouvez appliquer les jeux `write_set` et `read_set` comme suit :

```
# chmod A+user:bob:read_set/write_set:allow file.2
# ls -v file.2
-rw-r--r--+  1 root     root        2693 Jul 20 14:26 file.2
0:user:bob:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Configuration d'héritage d'ACL dans des fichiers ZFS en format détaillé

Vous pouvez déterminer comment les ACL sont héritées ou non dans les fichiers et répertoires. Par défaut, les ACL ne sont pas propagées. Si vous configurez une ACL non triviale dans un répertoire, aucun répertoire subséquent n'en hérite. Vous devez spécifier l'héritage d'une ACL dans un fichier ou un répertoire.

La propriété `aclinherit` peut être définie de manière globale pour un système de fichiers. Par défaut, `aclinherit` est défini sur `restricted`.

Pour plus d'informations, reportez-vous à la section “Héritage ACL” à la page 230.

EXEMPLE 7-6 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 `joe` dans le fichier `test.dir`.

```
# chmod A+user:joe:read_data/write_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:53 test.dir
0:user:joe:list_directory/read_data/add_file/write_data/execute:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
```

```
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Si un sous-répertoire `test.dir` est créé, l'ACE pour l'utilisateur `joe` n'est pas propagée. L'utilisateur `joe` n'aurait accès à `sub.dir` que si les autorisations de `sub.dir` lui accordaient un accès en tant que propriétaire de fichier, membre de groupe ou `everyone@`.

```
# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x   2 root     root           2 Jul 20 14:54 test.dir/sub.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

EXEMPLE 7-7 Attribution d'héritage d'ACL dans les fichiers et les répertoires

Cette série d'exemples identifie les ACE du fichier et du répertoire qui sont appliquées lorsque l'indicateur `file_inherit` est paramétré.

Dans cet exemple, les autorisations `read_data/write_data` sont ajoutées pour les fichiers dans le répertoire `test2.dir` pour l'utilisateur `joe` afin qu'il dispose de l'accès en lecture à tout nouveau fichier :

```
# chmod A+user:joe:read_data/write_data:file_inherit:allow test2.dir
# ls -dv test2.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:55 test2.dir
0:user:joe:read_data/write_data:file_inherit:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Dans l'exemple suivant, les autorisations de l'utilisateur `joe` sont appliquées au fichier `test2.dir/file.2` récemment créé. L'héritage d'ACL étant accordé (`read_data:file_inherit:allow`), l'utilisateur `joe` peut lire le contenu de tout nouveau fichier.

```
# touch test2.dir/file.2
# ls -v test2.dir/file.2
-rw-r--r--+  1 root     root           0 Jul 20 14:56 test2.dir/file.2
```

```
0:user:joe:read_data:inherited:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Dans la mesure où la propriété `aclinherit` pour ce système de fichiers est paramétrée sur le mode par défaut, `restricted`, l'utilisateur `joe` ne dispose pas de l'autorisation `write_data` pour le fichier `file.2` car l'autorisation de groupe du fichier ne le permet pas.

Notez que l'autorisation `inherit_only` appliquée lorsque les indicateurs `file_inherit` ou `dir_inherit` sont définis, est utilisée pour propager l'ACL dans la structure du répertoire. Ainsi, l'utilisateur `joe` se voit uniquement accorder ou refuser l'autorisation des autorisations `everyone@`, à moins qu'il ne soit le propriétaire du fichier ou membre du groupe propriétaire du fichier. Par exemple :

```
# mkdir test2.dir/subdir.2
# ls -dv test2.dir/subdir.2
drwxr-xr-x+  2 root     root           2 Jul 20 14:57 test2.dir/subdir.2
0:user:joe:list_directory/read_data/add_file/write_data:file_inherit
/inherit_only/inherited:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

La série d'exemples suivants identifie les ACL du fichier et du répertoire appliquées lorsque les indicateurs `file_inherit` et `dir_inherit` sont paramétrés.

Dans l'exemple suivant, l'utilisateur `joe` se voit accorder les droits de lecture, d'écriture et d'exécution hérités des fichiers et répertoires récemment créés.

```
# chmod A+user:joe:read_data/write_data/execute:file_inherit/dir_inherit:allow
test3.dir
# ls -dv test3.dir
drwxr-xr-x+  2 root     root           2 Jul 20 15:00 test3.dir
0:user:joe:list_directory/read_data/add_file/write_data/execute
:file_inherit/dir_inherit:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Le texte `inherited` de la sortie ci-dessous est un message d'information qui indique que l'ACE est héritée.

```
# touch test3.dir/file.3
# ls -v test3.dir/file.3
-rw-r--r--+  1 root     root           0 Jul 20 15:01 test3.dir/file.3
0:user:joe:read_data:inherited:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow

# touch test3.dir/file.3
# ls -v test3.dir/file.3
-rw-r--r--+  1 root     root           0 Jun 23 15:25 test3.dir/file.3
0:user:joe:read_data:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow

# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+  2 root     root           2 Jun 23 15:26 test3.dir/subdir.1
0:user:joe:list_directory/read_data/execute:file_inherit/dir_inherit
:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/read_attributes
/write_attributes/read_acl/write_acl/write_owner/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Dans les exemples ci-dessus, les bits d'autorisation du répertoire parent pour `group@` et `everyone@` n'accordent pas les autorisations d'écriture et d'exécution. Par conséquent, l'utilisateur `joe` se voit refuser ces autorisations. La propriété par défaut de `aclinherit` est `restricted`, ce qui signifie que les autorisations `write_data` et `execute` ne sont pas héritées.

Dans l'exemple suivant, l'utilisateur `joe` se voit accorder les autorisations de lecture, d'écriture et d'exécution qui sont héritées pour les fichiers récemment créés, mais elles ne sont pas propagées vers tout contenu subséquent du répertoire.

```
# chmod A+user:joe:read_data/write_data/execute:file_inherit/no_propagate:allow
test4.dir
# ls -dv test4.dir
drwxr--r--+  2 root     root           2 Mar  1 12:11 test4.dir
0:user:joe:list_directory/read_data/add_file/write_data/execute
:file_inherit/no_propagate:allow
```

```
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow
```

Comme l'exemple suivant l'illustre, les autorisations `read_data/write_data/execute` de l'utilisateur `joe` sont réduites en fonction des autorisations du groupe propriétaire.

```
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+  1 root     root           0 Jul 20 15:09 test4.dir/file.4
0:user:joe:read_data:inherited:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

EXEMPLE 7-8 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through

Si la propriété `aclinherit` du système de fichiers `tank/cindy` est définie sur `passthrough`, l'utilisateur `joe` hérite de l'ACL appliquée à `test4.dir` pour le nouveau fichier `file.5` de la manière suivante :

```
# zfs set aclinherit=passthrough tank/cindy
# touch test4.dir/file.5
# ls -v test4.dir/file.5
-rw-r--r--+  1 root     root           0 Jul 20 14:16 test4.dir/file.5
0:user:joe:read_data/write_data/execute:inherited:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

EXEMPLE 7-9 Héritage d'ACL avec mode d'héritage ACL défini sur Discard

Si la propriété `aclinherit` d'un système de fichiers est définie sur `discard`, il est alors possible de supprimer les ACL avec les bits d'autorisation lors d'un changement de répertoire. Par exemple :

```
# zfs set aclinherit=discard tank/cindy
# chmod A+user:joe:read_data/write_data/execute:dir_inherit:allow test5.dir
# ls -dv test5.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:18 test5.dir
```

```
0:user:joe:list_directory/read_data/add_file/write_data/execute
:dir_inherit:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Si vous décidez ultérieurement de renforcer les bits d'autorisation d'un répertoire, l'ACL non triviale est supprimée. Par exemple :

```
# chmod 744 test5.dir
# ls -dv test5.dir
drwxr--r--   2 root     root           2 Jul 20 14:18 test5.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow
```

EXEMPLE 7-10 Héritage d'ACL avec mode d'héritage de liste défini sur Noallow

Dans l'exemple suivant, deux ACL non triviales avec héritage de fichier sont définies. Une ACL accorde l'autorisation `read_data`, tandis qu'une autre refuse cette autorisation. Cet exemple illustre également comment spécifier deux ACE dans la même commande `chmod`.

```
# zfs set aclinherit=noallow tank/cindy
# chmod A+user:joe:read_data:file_inherit:deny,user:lp:read_data:file_inherit:allow
test6.dir
# ls -dv test6.dir
drwxr-xr-x+  2 root     root           2 Jul 20 14:22 test6.dir
0:user:joe:read_data:file_inherit:deny
1:user:lp:read_data:file_inherit:allow
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
3:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
4:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Comme l'illustre l'exemple suivant, lors de la création d'un nouveau fichier, l'ACL qui accorde l'autorisation `read_data` est supprimée.

```
# touch test6.dir/file.6
```

```
# ls -v test6.dir/file.6
-rw-r--r--+  1 root     root           0 Jul 20 14:23 test6.dir/file.6
0:user:joe:read_data:inherited:deny
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Configuration et affichage d'ACL dans des fichiers ZFS en format compact

Vous pouvez définir et afficher les autorisations relatives aux fichiers ZFS en format compact utilisant 14 lettres uniques pour représenter les autorisations. Les lettres représentant les autorisations compactes sont répertoriées dans Tableau 7-2, “Privilèges d'accès d'ACL” et Tableau 7-4, “Indicateurs d'héritage d'ACL”.

Vous pouvez afficher les listes d'ACL compactes pour les fichiers et les répertoires à l'aide de la commande `ls -V`. Par exemple :

```
# ls -V file.1
-rw-r--r--   1 root     root      206695 Jul 20 14:27 file.1
owner@:rw-p--aARWcCos:-------:allow
group@:r-----a-R-c--s:-------:allow
everyone@:r-----a-R-c--s:-------:allow
```

La sortie d'ACL compacte est décrite comme suit :

`owner@`
: Le propriétaire peut lire et modifier le contenu du fichier (`rw=read_data/write_data`), (`p=append_data`). Il peut également modifier les attributs du fichier tels que les horodatages, les attributs étendus et les ACL (`a=read_attributes`, `W=write_xattr`, `R=read_xattr`, `A=write_attributes`, `c=read_acl`, `C=write_acl`). De plus, le propriétaire peut modifier la propriété du fichier (`o=write_owner`).

  L'autorisation d'accès `synchronize` (`s`) n'est pas implémentée pour le moment.

`group@`
: Les autorisations de lecture du fichier (`r=read_data`) et de ses attributs (`a=read_attributes`, `R=read_xattr`, `c=read_acl`) sont attribuées au groupe.

  L'autorisation d'accès `synchronize` (`s`) n'est pas implémentée pour le moment.

`everyone@`	Les autorisations de lecture du fichier et de ses attributs (`r=read_data`, `a=append_data`, `R=read_xattr`, `c=read_acl` et `s=synchronize`). L'autorisation d'accès `synchronize` (`s`) n'est pas implémentée pour le moment.

Le format d'ACL compact dispose des avantages suivants par rapport au format d'ACL détaillé :

- Les autorisations peuvent être spécifiées en tant qu'arguments de position pour la commande `chmod`.
- Les tirets (-), qui n'identifient aucune autorisation, peuvent être supprimés. Seules les lettres nécessaires doivent être spécifiées.
- Les indicateurs d'autorisations et d'héritage sont configurés de la même manière.

Pour obtenir des informations sur l'utilisation du format d'ACL détaillé, consultez “Configuration et affichage d'ACL dans des fichiers ZFS en format détaillé” à la page 235.

EXEMPLE 7-11 Configuration et affichage des ACL en format compact

Dans l'exemple suivant, une ACL triviale existe dans le fichier `file.1` :

```
# ls -V file.1
-rw-r--r--   1 root     root      206695 Jul 20 14:27 file.1
owner@:rw-p--aARWcCos:-------:allow
group@:r-----a-R-c--s:-------:allow
everyone@:r-----a-R-c--s:-------:allow
```

Dans cet exemple, les autorisations `read_data/execute` sont ajoutées à l'utilisateur `gozer` sur le fichier `file.1`.

```
# chmod A+user:joe:rx:allow file.1
# ls -V file.1
-rw-r--r--+  1 root     root      206695 Jul 20 14:27 file.1
user:joe:r-x-----------:-------:allow
owner@:rw-p--aARWcCos:-------:allow
group@:r-----a-R-c--s:-------:allow
everyone@:r-----a-R-c--s:-------:allow
```

Dans l'exemple suivant, l'utilisateur `joe` se voit accorder les droits de lecture, d'écriture et d'exécution qui sont hérités des fichiers et répertoires récemment créés grâce à l'utilisation de l'ACL compacte.

```
# chmod A+user:joe:rwx:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+  2 root     root           2 Jul 20 14:33 dir.2
user:joe:rwx-----------:fd-----:allow
owner@:rwxp-DaARWcCos:-------:allow
group@:r-x---a-R-c--s:-------:allow
```

```
everyone@:r-x---a-R-c--s:-------:allow
```

Vous pouvez également couper et coller les autorisations et les indicateurs d'héritage à partir de la sortie `ls-V` en format `chmod` compact. Par exemple, pour dupliquer les autorisations et les indicateurs d'héritage du fichier `dir.2` de l'utilisateur `joe` à l'utilisateur `cindy` dans le fichier `dir.2`, copiez et collez les autorisations et les indicateurs d'héritage (`rwx-----------:f-----:allow`) dans la commande `chmod`. Par exemple :

```
# chmod A+user:cindy:rwx-----------:fd-----:allow dir.2
# ls -dV dir.2
drwxr-xr-x+  2 root     root           2 Jul 20 14:33 dir.2
user:cindy:rwx-----------:fd-----:allow
user:joe:rwx-----------:fd-----:allow
owner@:rwxp-DaARWcCos:-------:allow
group@:r-x---a-R-c--s:-------:allow
everyone@:r-x---a-R-c--s:-------:allow
```

EXEMPLE 7-12 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through

Un système de fichiers dont la propriété `aclinherit` est définie sur `passthrough` hérite de toutes les entrées d'ACL pouvant être héritées, sans qu'aucune modification ne leur soit apportée. Lorsque cette propriété est définie sur `passthrough`, les fichiers sont créés avec un mode d'autorisation déterminé par les ACE pouvant être héritées. Si aucune ACE pouvant être héritée n'affecte le mode d'autorisation, ce mode est alors défini en fonction du mode demandé à partir de l'application.

Les exemples suivants respectent la syntaxe ACL compacte pour illustrer le processus d'héritage des bits d'autorisation en définissant le mode `aclinherit` sur la valeur `passthrough` .

Dans cet exemple, une ACL est définie sur `test1.dir` pour forcer l'héritage. La syntaxe crée une entrée d'ACL `owner@`, `group@` et `everyone@` pour les fichiers nouvellement créés. Les répertoires nouvellement créés héritent d'une entrée d'ACL `@owner`, `group@` et `everyone@`.

```
# zfs set aclinherit=passthrough tank/cindy
# pwd
/tank/cindy
# mkdir test1.dir

# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,
everyone@::fd:allow test1.dir
# ls -Vd test1.dir
drwxrwx---+  2 root     root           2 Jul 20 14:42 test1.dir
owner@:rwxpdDaARWcCos:fd-----:allow
group@:rwxp----------:fd-----:allow
everyone@:----------:fd-----:allow
```

Dans cet exemple, un fichier nouvellement créé hérite de l'ACL dont les fichiers nouvellement créés doivent hériter d'après ce qui a été spécifié.

```
# cd test1.dir
# touch file.1
# ls -V file.1
-rwxrwx---+  1 root     root           0 Jul 20 14:44 file.1
owner@:rwxpdDaARWcCos:------I:allow
group@:rwxp----------:------I:allow
everyone@:--------------:------I:allow
```

Dans cet exemple, un répertoire nouvellement créé hérite à la fois des ACE contrôlant l'accès à ce répertoire et des ACE à appliquer ultérieurement aux enfants de ce répertoire.

```
# mkdir subdir.1
# ls -dV subdir.1
drwxrwx---+  2 root     root           2 Jul 20 14:45 subdir.1
owner@:rwxpdDaARWcCos:fd----I:allow
group@:rwxp----------:fd----I:allow
everyone@:--------------:fd----I:allow
```

Les entrées `fd----I` servent à propager l'héritage et ne sont pas prises en compte durant le contrôle d'accès.

Dans l'exemple suivant, un fichier est créé à l'aide d'une ACL triviale dans un autre répertoire où les ACE héritées ne sont pas présentes.

```
# cd /tank/cindy
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r--   1 root     root           0 Jul 20 14:48 file.2
owner@:rw-p--aARWcCos:-------:allow
group@:r-----a-R-c--s:-------:allow
everyone@:r-----a-R-c--s:-------:allow
```

EXEMPLE 7-13 Héritage d'ACL avec mode d'héritage ACL défini sur Pass Through-X

Lorsque `aclinherit=passthrough-x` est activé, les fichiers sont créés avec l'autorisation d'exécution (x) pour `propriétaire@`, `groupe@` ou `tous les utilisateurs@`, mais seulement si l'autorisation d'exécution est définie dans le mode de création de fichier et dans une ACE héritable qui affecte le mode.

L'exemple suivant montre comment hériter l'autorisation d'exécution en définissant le mode `aclinherit` sur `passthrough-x`.

```
# zfs set aclinherit=passthrough-x tank/cindy
```

L'ACL suivante est définie sur `/tank/cindy/test1.dir` pour permettre l'héritage des ACL exécutables pour les fichiers de `owner@`.

```
# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,
```

```
everyone@::fd:allow test1.dir
# ls -Vd test1.dir
drwxrwx---+  2 root     root           2 Jul 20 14:50 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 Jul 20 14:52 test1.dir/file1
owner@:rw-pdDaARWcCos:------I:allow
group@:rw-p----------:------I:allow
everyone@:----------:------I:allow
```

Ensuite, un fichier exécutable appelé `t` est généré à l'aide du compilateur `cc` dans le répertoire `testdir`.

```
# cc -o t t.c
# ls -V t
-rwxrwx---+  1 root     root        7396 Dec  3 15:19 t
owner@:rwxpdDaARWcCos:------I:allow
group@:rwxp----------:------I:allow
everyone@:----------:------I:allow
```

Les autorisations obtenues sont `0770` car `cc` a demandé des autorisations `0777`, ce qui a entraîné l'héritage de l'autorisation d'exécution à partir des entrées `propriétaire@`, `groupe@` et `tous les utilisateurs@`.

EXEMPLE 7-14 Interactions entre les ACL et les opérations `chmod` sur les fichiers ZFS

Les exemples suivants illustrent l'incidence de certaines valeurs des propriétés `aclmode` et `aclinherit` sur l'interaction des ACL existantes avec une opération `chmod` modifiant les autorisations de répertoire ou de fichier en vue de restreindre ou d'augmenter les autorisations d'ACL existantes à des fins de conformité avec le groupe propriétaire.

Dans cet exemple, la propriété `aclmode` est définie sur `mask` et la propriété `aclinherit` sur `restricted`. Les autorisations d'ACL sont affichées en mode compact dans cet exemple, ce qui permet de mieux repérer les modifications apportées aux autorisations.

Paramètres de propriété du fichier et des groupes et autorisations d'ACL initiaux :

```
# zfs set aclmode=mask pond/whoville
# zfs set aclinherit=restricted pond/whoville

# ls -lV file.1
```

```
-rwxrwx---+  1 root     root      206695 Aug 30 16:03 file.1
user:amy:r-----a-R-c---:-------:allow
user:rory:r-----a-R-c---:-------:allow
group:sysadmin:rw-p--aARWc---:-------:allow
group:staff:rw-p--aARWc---:-------:allow
owner@:rwxp--aARWcCos:-------:allow
group@:rwxp--aARWc--s:-------:allow
everyone@:------a-R-c--s:-------:allow
```

Une opération `chown` modifie la propriété du fichier `file.1`, et la sortie est visible par l'utilisateur propriétaire, `amy`. Par exemple :

```
# chown amy:staff file.1
# su - amy
$ ls -lV file.1
-rwxrwx---+  1 amy      staff     206695 Aug 30 16:03 file.1
user:amy:r-----a-R-c---:-------:allow
user:rory:r-----a-R-c---:-------:allow
group:sysadmin:rw-p--aARWc---:-------:allow
group:staff:rw-p--aARWc---:-------:allow
owner@:rwxp--aARWcCos:-------:allow
group@:rwxp--aARWc--s:-------:allow
everyone@:------a-R-c--s:-------:allow
```

Les opérations `chmod` suivantes font passer les autorisations à un mode plus restrictif. Dans cet exemple, les autorisations d'ACL modifiées du groupe `sysadmin` et du groupe `staff` n'excèdent pas les autorisations du groupe propriétaire.

```
$ chmod 640 file.1
$ ls -lV file.1
-rw-r-----+  1 amy      staff     206695 Aug 30 16:03 file.1
user:amy:r-----a-R-c---:-------:allow
user:rory:r-----a-R-c---:-------:allow
group:sysadmin:r-----a-R-c---:-------:allow
group:staff:r-----a-R-c---:-------:allow
owner@:rw-p--aARWcCos:-------:allow
group@:r-----a-R-c--s:-------:allow
everyone@:------a-R-c--s:-------:allow
```

L'opération `chmod` suivante fait passer les autorisations à un mode moins restrictif. Dans cet exemple, les autorisations d'ACL modifiées du groupe `sysadmin` et du groupe `staff` sont restaurées pour accorder les mêmes autorisations que celles du groupe propriétaire.

```
$ chmod 770 file.1
$ ls -lV file.1
-rwxrwx---+  1 amy      staff     206695 Aug 30 16:03 file.1
user:amy:r-----a-R-c---:-------:allow
user:rory:r-----a-R-c---:-------:allow
group:sysadmin:rw-p--aARWc---:-------:allow
group:staff:rw-p--aARWc---:-------:allow
owner@:rwxp--aARWcCos:-------:allow
group@:rwxp--aARWc--s:-------:allow
everyone@:------a-R-c--s:-------:allow
```

Application d'attributs spéciaux aux fichiers ZFS

Les exemples suivants montrent comment appliquer et afficher des attributs spéciaux, tels que l'immuabilité ou l'accès en lecture seule, à des fichiers ZFS.

Pour plus d'informations sur l'affichage et l'application d'attributs spéciaux, reportez-vous aux pages de manuel `ls(1)` et `chmod(1)`.

EXEMPLE 7-15 Application de l'immuabilité à un fichier ZFS

Respectez la syntaxe suivante pour rendre un fichier immuable :

```
# chmod S+ci file.1
# echo this >>file.1
-bash: file.1: Not owner
# rm file.1
rm: cannot remove `file.1': Not owner
```

Vous pouvez afficher les attributs spéciaux qui s'appliquent à des fichiers ZFS en respectant la syntaxe suivante :

```
# ls -l/c file.1
-rw-r--r--+  1 root     root      206695 Jul 20 14:27 file.1
{A-----im----}
```

Respectez la syntaxe suivante pour annuler l'immuabilité d'un fichier :

```
# chmod S-ci file.1
# ls -l/c file.1
-rw-r--r--+  1 root     root      206695 Jul 20 14:27 file.1
{A------m----}
# rm file.1
```

EXEMPLE 7-16 Application d'un accès en lecture seule à un fichier ZFS

L'exemple suivant indique comment appliquer l'accès en lecture seule à un fichier ZFS.

```
# chmod S+cR file.2
# echo this >>file.2
-bash: file.2: Not owner
```

EXEMPLE 7-17 Affichage et modification des attributs d'un fichier ZFS

Vous pouvez afficher et définir des attributs spéciaux en respectant la syntaxe suivante :

```
# ls -l/v file.3
-r--r--r--   1 root     root      206695 Jul 20 14:59 file.3
{archive,nohidden,noreadonly,nosystem,noappendonly,nonodump,
noimmutable,av modified,noav_quarantined,nonounlink,nooffline,nosparse}
```

```
# chmod S+cR file.3
# ls -l/v file.3
-r--r--r--   1 root     root      206695 Jul 20 14:59 file.3
{archive,nohidden,readonly,nosystem,noappendonly,nonodump,noimmutable,
av_modified,noav_quarantined,nonounlink,nooffline,nosparse}
```

Certains de ces attributs s'appliquent uniquement à un environnement Oracle Solaris SMB.

Vous pouvez effacer tous les attributs d'un fichier. Par exemple :

```
# chmod S-a file.3
# ls -l/v file.3
-r--r--r--   1 root     root      206695 Jul 20 14:59 file.3
{noarchive,nohidden,noreadonly,nosystem,noappendonly,nonodump,
noimmutable,noav_modified,noav_quarantined,nonounlink,nooffline,nosparse}
```

♦ ♦ ♦ CHAPITRE 8

Administration déléguée de ZFS dans Oracle Solaris

Ce chapitre explique comment utiliser les fonctions d'administration déléguée pour permettre aux utilisateurs ne disposant pas des autorisations nécessaires d'effectuer des tâches d'administration de ZFS.

Ce chapitre contient les sections suivantes :

Présentation de l'administration déléguée de ZFS

L'administration déléguée de ZFS vous permet de distribuer des autorisations précises à des utilisateurs ou à des groupes spécifiques, voire à tous les utilisateurs. Deux types d'autorisations déléguées sont pris en charge :

- Les autorisations individuelles suivantes peuvent être explicitement déléguées : autorisation de création (`create`), autorisation de destruction (`destroy`), autorisation de montage (`mount`), autorisation de créer des instantanés (`snapshot`), etc.
- Des groupes d'autorisations appelés *jeux d'autorisations* peuvent être définis. Si un jeu d'autorisations est modifié, tout utilisateur de ce jeu de d'autorisations est automatiquement affecté par ces modifications. Les jeux d'autorisations commencent par le symbole `@` et sont limités à 64 caractères. Quand le symbole `@` a été supprimé, les autres caractères du même nom de jeu ont les mêmes restrictions que les noms de système de fichiers ZFS.

L'administration déléguée de ZFS offre des fonctions similaires au modèle de sécurité RBAC. La délégation ZFS offre les avantages suivants pour la gestion des pools de stockage et systèmes de fichiers ZFS :

- Les autorisations sont transférées avec le pool de stockage ZFS lorsqu'un pool est migré.

- L'héritage dynamique vous permet de contrôler la propagation des autorisations dans les systèmes de fichiers.
- Il est possible de définir une configuration de manière à ce que seul le créateur d'un système de fichiers puisse détruire celui-ci.
- Les autorisations peuvent être déléguées à des systèmes de fichiers spécifiques. Tout nouveau système de fichiers peut automatiquement récupérer des autorisations.
- Simplifie l'administration de systèmes de fichiers en réseau (NFS, Network File System). Un utilisateur disposant d'autorisations explicites peut, par exemple, créer un instantané sur un système NFS dans le répertoire `.zfs/snapshot` approprié.

Considérez l'utilisation de l'administration déléguée pour la répartition des tâches ZFS. Pour plus d'informations sur l'utilisation de RBAC pour gérer les tâches d'administration Oracle Solaris générales, reportez-vous au Chapitre 1, “ A propos de l’utilisation de droits pour contrôle Users and Processes ” du manuel “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ”.

Désactivation des droits délégués de ZFS

La propriété `delegation` du pool permet de contrôler les fonctions d'administration déléguées. Par exemple :

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

Par défaut, la propriété `delegation` est activée.

Délégation d'autorisations ZFS

Vous pouvez utiliser la commande `zfs allow` pour déléguer les autorisations sur les systèmes de fichiers ZFS vers des utilisateurs non root en utilisant l'une des méthodes suivantes :

- Vous pouvez déléguer des autorisations individuelles à un utilisateur, à un groupe, voire à tous les utilisateurs.
- Vous pouvez déléguer des groupes d'autorisations individuelles sous forme de *jeu d'autorisations* à un utilisateur, à un groupe, voire à tous les utilisateurs.
- Vous pouvez déléguer des autorisations localement, soit uniquement au système de fichiers actuel, soit à l'ensemble de ses descendants.

Le tableau suivant décrit les opérations pouvant être déléguées et toute autorisation dépendante requise pour réaliser ces opérations déléguées.

Autorisation (sous-commande)	Description	Dépendances
`allow`	Autorisation d'accorder des autorisations qui vous ont été octroyées à un autre utilisateur.	Nécessité de disposer de l'autorisation en cours d'octroi.
`clone`	Autorisation de cloner tout instantané du jeu de données.	Nécessité de disposer également des autorisations `mount` et `promote` sur le système de fichiers d'origine.
`créer`	Autorisation de créer des jeux de données descendants.	Nécessité de disposer de l'autorisation `mount`.
`destroy`	Autorisation de détruire un jeu de données.	Nécessité de disposer de l'autorisation `mount`.
`diff`	Autorisation d'identifier les chemins d'accès à l'intérieur d'un jeu de données.	Les utilisateurs non root ont besoin de cette autorisation pour utiliser la commande `zfs diff`.
`hold`	Autorisation de conservation d'un instantané.	
`mount`	Autorisation de monter et démonter un système de fichiers, et de créer et détruire les liens vers des périphériques de volume.	
`promouvoir`	Autorisation de promouvoir le clonage d'un jeu de données.	Nécessité de disposer également des autorisations `mount` et `promote` sur le système de fichiers d'origine.
`recevoir`	Autorisation de créer des systèmes de fichiers descendants à l'aide de la commande `zfs receive`.	Nécessité de disposer également des autorisations `mount` et `create`.
`version`	Autorisation de libérer un instantané conservé, ce qui peut détruire l'instantané.	
`renommer`	Autorisation de renommer un jeu de données.	Nécessité de disposer également des autorisations `create` et `mount` sur le nouveau parent.
`rollback`	Autorisation de restaurer un instantané.	
`send`	Autorisation d'envoyer un flux d'instantané.	
`share`	Autorisation de partager et de départager un système de fichiers	`share` et `share.nfs` sont tous les deux nécessaires pour créer un partage NFS. `share` et `share.smb` sont tous les deux nécessaires pour créer un partage SMB.
`snapshot`	Autorisation de créer un instantané d'un jeu de données.	

Vous pouvez déléguer le jeu d'autorisations suivant mais une autorisation peut être limitée à l'accès, à la lecture ou à la modification :

- `groupquota`

- groupused
- key
- keychange
- userprop
- userquota
- userused

Vous pouvez en outre déléguer l'administration des propriétés ZFS suivantes à des utilisateurs non root :

- aclinherit
- aclmode
- atime
- canmount
- casesensitivity
- checksum
- compression
- copies
- dedup
- defaultgroupquota
- defaultuserquota
- devices
- encryption
- exec
- keysource
- logbias
- mountpoint
- nbmand
- normalization
- primarycache
- quota
- readonly
- recordsize
- refquota
- refreservation
- reservation
- rstchown

- secondarycache
- setuid
- shadow
- share.nfs
- share.smb
- snapdir
- sync
- utf8only
- version
- volblocksize
- volsize
- vscan
- xattr
- zoned

Certaines de ces propriétés ne peuvent être définies qu'à la création d'un jeu de données. Pour une description de ces propriétés, reportez-vous à la section “Présentation des propriétés ZFS” à la page 139.

Délégation des autorisations ZFS (`zfs allow`)

La syntaxe `zfs allow` suivante :

```
zfs allow -[ldugecs] everyone|user|group[,...] perm|@setname,...] filesystem| volume
```

La syntaxe de `zfs allow` suivante (en gras) identifie les utilisateurs auxquels les autorisations sont déléguées :

`zfs allow [-uge]|user|group|everyone` `[,...]` *filesystem* | *volume*

Vous pouvez spécifier plusieurs entrées sous forme de liste séparée par des virgules. Si aucune option `-uge` n'est spécifiée, l'argument est interprété en premier comme le mot-clé `everyone`, puis comme un nom d'utilisateur et enfin, comme un nom de groupe. Pour spécifier un utilisateur ou un groupe nommé "everyone", utilisez l'option `-u` ou l'option `-g`. Pour spécifier un groupe portant le même nom qu'un utilisateur, utilisez l'option `-g`. L'option `-c` sert à déléguer des autorisations create-time.

La syntaxe de `zfs allow` suivante (en gras) identifie la méthode de spécification des autorisations et jeux d'autorisations :

zfs allow [-s] ... perm|@setname [,...] *filesystem* | *volume*

Vous pouvez spécifier plusieurs autorisations sous forme de liste séparée par des virgules. Les noms d'autorisations sont identiques aux sous-commandes et propriétés ZFS. Pour plus d'informations, reportez-vous à la section précédente.

Les autorisations peuvent être regroupées en *jeux d'autorisations* et sont identifiées par l'option `-s`. Les jeux d'autorisations peuvent être utilisés par d'autres commandes `zfs allow` pour le système de fichiers spécifié et ses descendants. Les jeux d'autorisations sont évalués dynamiquement et de ce fait, toute modification apportée à un jeu est immédiatement mise à jour. Les jeux d'autorisations doivent se conformer aux mêmes critères d'attribution de noms que les systèmes de fichiers ZFS, à ceci près que leurs noms doivent commencer par le caractère arobase (`@`) et ne pas dépasser 64 caractères.

La syntaxe de `zfs allow` suivante (en gras) identifie la méthode de délégation des autorisations :

zfs allow [-ld] ... ... *filesystem* | *volume*

L'option `-l` indique que les autorisations sont accordées au système de fichiers spécifié mais pas à ses descendants, à moins de spécifier également l'option `-d`. L'option `-d` indique que les autorisations sont accordées aux systèmes de fichiers descendants mais pas à ce système de fichiers, à moins de spécifier également l'option `- l`. Si aucune option n'est spécifiée, les autorisations sont accordées au système de fichiers ou au volume ainsi qu'à ses descendants.

Suppression des autorisations déléguées de ZFS (`zfs unallow`)

Vous pouvez supprimer des autorisations précédemment déléguées à l'aide de la commande `zfs unallow`.

Par exemple, supposons que vous déléguiez les autorisations `create`, `destroy`, `mount` et `snapshot` comme suit :

```
# zfs allow cindy create,destroy,mount,snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy --------------------------------
Local+Descendent permissions:
user cindy create,destroy,mount,snapshot
```

Pour supprimer ces autorisations, vous devez respecter la syntaxe suivante :

```
# zfs unallow cindy tank/home/cindy
# zfs allow tank/home/cindy
```

Délégation d'autorisations ZFS (exemples)

EXEMPLE 8-1 Délégation d'autorisations à un utilisateur individuel

Lorsque vous déléguez les autorisations `create` et `mount` à un utilisateur individuel, vous devez vous assurer qu'il dispose d'un droit d'accès au de point de montage sous-jacent.

Par exemple, pour déléguer à l'utilisateur `mark` les autorisations `create` et `mount` sur le système de fichiers `tank`, définissez au préalable ces autorisations :

```
# chmod A+user:mark:add_subdirectory:fd:allow /tank/home
```

Utilisez ensuite la commande `zfs allow` pour déléguer les autorisations `create`, `destroy` et `mount`. Par exemple :

```
# zfs allow mark create,destroy,mount tank/home
```

L'utilisateur `mark` peut dorénavant créer ses propres systèmes de fichiers dans le système de fichiers `tank/home`. Par exemple :

```
# su mark
mark$ zfs create tank/home/mark
mark$ ^D
# su lp
$ zfs create tank/home/lp
cannot create 'tank/home/lp': permission denied
```

EXEMPLE 8-2 Délégation des autorisations de création (`create`) et de destruction (`destroy`) à un groupe

L'exemple suivant décrit comment configurer un système de fichiers de manière à ce que tout membre du groupe `staff` puisse créer et monter des systèmes de fichiers dans le système de fichiers `tank/home`, et détruire ses propres systèmes de fichiers. Toutefois, les membres du groupe `staff` ne sont pas autorisés à détruire les systèmes de fichiers des autres utilisateurs.

```
# zfs allow staff create,mount tank/home
# zfs allow -c create,destroy tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
create,destroy
Local+Descendent permissions:
group staff create,mount
# su cindy
cindy% zfs create tank/home/cindy/files
```

```
cindy% exit
# su mark
mark% zfs create tank/home/mark/data
mark% exit
cindy% zfs destroy tank/home/mark/data
cannot destroy 'tank/home/mark/data': permission denied
```

EXEMPLE 8-3 Délégation d'autorisations au niveau approprié d'un système de fichiers

Assurez-vous de déléguer les autorisations aux utilisateurs au niveau approprié du système de fichiers. Par exemple, l'utilisateur `mark` est autorisé à`create`, `destroy` et `mount` les autorisations pour les systèmes de fichiers locaux et descendants. L'autorisation locale de créer un instantané du système de fichiers `tank/home` a été délégué à l'utilisateur `mark`, mais pas celle de créer un instantané de son propre système de fichiers. L'autorisation `snapshot` ne lui a donc pas été déléguée au niveau approprié du système de fichiers.

```
# zfs allow -l mark snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
create,destroy
Local permissions:
user mark snapshot
Local+Descendent permissions:
group staff create,mount
# su mark
mark$ zfs snapshot tank/home@snap1
mark$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

Pour déléguer à l'utilisateur `mark` cette autorisation au niveau du système de fichiers descendants, utilisez l'option `zfs allow -d`. Par exemple :

```
# zfs unallow -l mark snapshot tank/home
# zfs allow -d mark snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
create,destroy
Descendent permissions:
user mark snapshot
Local+Descendent permissions:
group staff create,mount
# su mark
$ zfs snapshot tank/home@snap2
cannot create snapshot 'tank/home@snap2': permission denied
$ zfs snapshot tank/home/mark@snappy
```

L'utilisateur `mark` ne peut maintenant créer un instantané qu'à un niveau inférieur du système de fichiers `tank/home`.

EXEMPLE 8-4 Définition et utilisation d'autorisations déléguées complexes

Vous pouvez déléguer des autorisations spécifiques à des utilisateurs ou des groupes. Par exemple, la commande `zfs allow` suivante délègue des autorisations spécifiques au groupe `staff`. En outre, les autorisations `destroy` et `snapshot` sont déléguées après la création de systèmes de fichiers `tank/home`.

```
# zfs allow staff create,mount tank/home
# zfs allow -c destroy,snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home --------------------------------------
Create time permissions:
create,destroy,snapshot
Local+Descendent permissions:
group staff create,mount
```

Etant donné que l'utilisateur `mark` est membre du groupe `staff`, il peut créer des systèmes de fichiers dans `tank/home`. En outre, l'utilisateur `mark` peut créer un instantané de `tank/home/mark2` parce qu'il dispose des autorisations spécifiques pour le faire. Par exemple :

```
# su mark
$ zfs create tank/home/mark2
$ zfs allow tank/home/mark2
---- Permissions on tank/home/mark2 ---------------------------------
Local permissions:
user mark create,destroy,snapshot
---- Permissions on tank/home --------------------------------------
Create time permissions:
create,destroy,snapshot
Local+Descendent permissions:
group staff create,mount
```

L'utilisateur `mark` ne peut pas créer d'instantané dans `tank/home/mark` parce qu'il ne dispose pas des autorisations spécifiques pour le faire. Par exemple :

```
$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

Dans cet exemple, l'utilisateur `mark` possède l'autorisation `create` dans son répertoire d'accueil, ce qui signifie qu'il peut créer des instantanés. Ce scénario s'avère utile lorsque votre système de fichiers est monté sur un système NFS.

```
$ cd /tank/home/mark2
$ ls
$ cd .zfs
$ ls
shares snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x   2 mark    staff          2 Sep 27 15:55 snap1
```

```
$ pwd
/tank/home/mark2/.zfs/snapshot
$ mkdir snap2
$ zfs list
# zfs list -r tank/home
NAME                    USED  AVAIL  REFER  MOUNTPOINT
tank/home/mark           63K  62.3G    32K  /tank/home/mark
tank/home/mark2          49K  62.3G    31K  /tank/home/mark2
tank/home/mark2@snap1    18K      -    31K  -
tank/home/mark2@snap2      0      -    31K  -
$ ls
snap1  snap2
$ rmdir snap2
$ ls
snap1
```

EXEMPLE 8-5 Définition et utilisation d'un jeu d'autorisations délégué ZFS

L'exemple suivant décrit comment créer un jeu d'autorisations intitulé `@myset` et déléguer ce jeu d'autorisations ainsi que l'autorisation `rename` au groupe `staff` pour le système de fichiers `tank`. L'utilisateur `cindy`, membre du groupe `staff`, a l'autorisation de créer un système de fichiers dans `tank`. Par contre, l'utilisateur `lp` ne dispose pas de cette autorisation de création de systèmes de fichiers dans `tank`.

```
# zfs allow -s @myset create,destroy,mount,snapshot,promote,clone,readonly tank
# zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
@myset clone,create,destroy,mount,promote,readonly,snapshot
# zfs allow staff @myset,rename tank
# zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
@myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions:
group staff @myset,rename
# chmod A+group:staff:add_subdirectory:fd:allow tank
# su cindy
cindy% zfs create tank/data
cindy% zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
@myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions:
group staff @myset,rename
cindy% ls -l /tank
total 15
drwxr-xr-x   2 cindy   staff          2 Jun 24 10:55 data
cindy% exit
# su lp
$ zfs create tank/lp
```

```
cannot create 'tank/lp': permission denied
```

Affichage des autorisations ZFS déléguées (exemples)

Vous pouvez utiliser la commande suivante pour afficher les droits:

```
# zfs allow dataset
```

Cette commande affiche les autorisations définies ou accordées au jeu de données spécifié. La sortie contient les composants suivants :

- Jeux d'autorisations
- Autorisations individuelles ou autorisations à la création
- Jeu de données local
- Jeux de données locaux et descendants
- Jeux de données descendants uniquement

EXEMPLE 8-6 Affichage des autorisations d'administration déléguées de base

La sortie suivante indique que l'utilisateur `cindy` dispose des autorisations d'instantané `create`, `destroy`, `mount` sur le système de fichiers `tank/cindy`.

```
# zfs allow tank/cindy
-------------------------------------------------------------
Local+Descendent permissions on (tank/cindy)
user cindy create,destroy,mount,snapshot
```

EXEMPLE 8-7 Affichage des autorisations d'administration déléguée complexes

La sortie de cet exemple indique les autorisations suivantes sur les systèmes de fichiers `pool/fred` et `pool`.

Pour le système de fichiers `pool/fred` :

- Deux jeux d'autorisations sont définis :
  - `@eng` (`créer`, `détruire`, `instantané`, `montage`, `clone`, `promouvoir`, `renommer`)
  - `@simple` (`créer`, `monter`)
- Les autorisations à la création sont définies pour le jeu d'autorisations `@eng` et la propriété `mountpoint`. "A la création" signifie qu'une fois qu'un jeu de systèmes de fichiers est créé, le jeu d'autorisations `@eng` et l'autorisation de définir la propriété `mountpoint` sont déléguées.
- Le jeu d'autorisations `@eng` est délégué à l'utilisateur `tom`, et les autorisations `create`, `destroy` et `mount` pour les systèmes de fichiers locaux sont déléguées à l'utilisateur `joe`.

- Le jeu d'autorisations `@basic` ainsi que les autorisations `share` et `rename` pour les systèmes de fichiers locaux et descendants sont délégués à l'utilisateur `fred`.
- Le jeu d'autorisations `@basic` pour les systèmes de fichiers descendants uniquement est délégué à l'utilisateur `barney` et au groupe `staff`.

Pour le système de fichiers `pool` :

- Le jeu d'autorisations `@simple` (create, destroy, mount) est défini.
- Le jeu d'autorisations sur le système de fichiers local `@simple` est accordé au groupe `staff`.

La sortie de cet exemple est la suivante :

```
$ zfs allow pool/fred
---- Permissions on pool/fred --------------------------------------
Permission sets:
@eng create,destroy,snapshot,mount,clone,promote,rename
@simple create,mount
Create time permissions:
@eng,mountpoint
Local permissions:
user tom @eng
user joe create,destroy,mount
Local+Descendent permissions:
user fred @basic,share,rename
user barney @basic
group staff @basic
---- Permissions on pool -------------------------------------------
Permission sets:
@simple create,destroy,mount
Local permissions:
group staff @simple
```

Supprimer des autorisations déléguées (exemples)

Vous pouvez utiliser la commande `zfs unallow` pour supprimer des autorisations déléguées. Par exemple, l'utilisateur `cindy` possède les autorisations `create`, `destroy`, `mount` et `snapshot` sur le système de fichiers `tank/cindy`.

```
# zfs allow cindy create,destroy,mount,snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy --------------------------------
Local+Descendent permissions:
user cindy create,destroy,mount,snapshot
```

La syntaxe suivante de la commande `zfs unallow` supprime l'autorisation de réaliser des instantanés (`snapshot`) du système de fichiers `tank/home/cindy` accordée à l'utilisateur `cindy` :

```
# zfs unallow cindy snapshot tank/home/cindy
```

```
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy ---------------------------------
Local+Descendent permissions:
user cindy create,destroy,mount
cindy% zfs create tank/home/cindy/data
cindy% zfs snapshot tank/home/cindy@today
cannot create snapshot 'tank/home/cindy@today': permission denied
```

Autre exemple : l'utilisateur `mark` possède les autorisations suivantes sur le système de fichiers `tank/home/mark` :

```
# zfs allow tank/home/mark
---- Permissions on tank/home/mark ---------------------------------
Local+Descendent permissions:
user mark create,destroy,mount
-------------------------------------------------------------
```

La syntaxe suivante de la commande `zfs unallow` supprime toutes les autorisations accordées à l'utilisateur `mark` pour le système de fichiers `tank/home/mark` :

```
# zfs unallow mark tank/home/mark
```

La syntaxe suivante de la commande `zfs unallow` supprime un jeu d'autorisations sur le système de fichiers `tank`.

```
# zfs allow tank
---- Permissions on tank --------------------------------------------
Permission sets:
@myset clone,create,destroy,mount,promote,readonly,snapshot
Create time permissions:
create,destroy,mount
Local+Descendent permissions:
group staff create,mount
# zfs unallow -s @myset tank
# zfs allow tank
---- Permissions on tank --------------------------------------------
Create time permissions:
create,destroy,mount
Local+Descendent permissions:
group staff create,mount
```

♦ ♦ ♦ CHAPITRE 9

Rubriques avancées Oracle Solaris ZFS

Ce chapitre décrit les volumes ZFS, l'utilisation de ZFS dans un système Solaris avec zones installées, les pools root de remplacement ZFS et les profils de droits ZFS.

Ce chapitre contient les sections suivantes :

Volumes ZFS

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 endommagées. Vous devez faire preuve de prudence lors de la modification de la taille du volume.

En outre, si vous créez un instantané pour un volume modifiant la taille de ce dernier, cela peut provoquer des incohérences lorsque vous tentez d'annuler (roll back) l'instantané ou de créer un clone à partir de l'instantané.

Pour de plus amples informations concernant les propriétés de systèmes de fichiers applicables aux volumes, reportez-vous au Tableau 5-1, "Description des propriétés ZFS natives".

Pour afficher les informations de propriétés de volumes ZFS à l'aide de la commande `zfs get` ou `zfs get all`. Par exemple :

```
# zfs get all tank/vol
```

Si un point d'interrogation (?) s'affiche dans `volsize` de la sortie `zfs get`, cela signifie qu'une valeur est inconnue car une erreur d'E/S s'est produite. Par exemple :

```
# zfs get -H volsize tank/vol
tank/vol        volsize ?       local
```

Une erreur d'E/S indique généralement un problème lié à un périphérique de pool. Pour plus d'informations sur la résolution des problèmes liés aux périphériques, consultez “Identification des problèmes avec les pools de stockage ZFS” à la page 286.

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 275.

Utilisation d'un volume ZFS en tant que périphérique de swap ou de vidage

Lors de l'installation d'un système de fichiers root ZFS ou d'une migration à partir d'un système de fichiers root UFS, un périphérique de swap est créé sur un volume ZFS du pool root ZFS. Par exemple :

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

Lors de l'installation d'un système de fichiers root ZFS ou d'une migration à partir d'un système de fichiers root UFS, un périphérique de vidage est créé sur un volume ZFS du pool root ZFS. Le périphérique de vidage ne nécessite aucune administration une fois configuré. Par exemple :

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

Si vous devez modifier votre zone de swap ou votre périphérique de vidage après l'installation du système, utilisez les commandes `swap` et `dumpadm` de la même manière que dans les versions précédentes de Solaris. Si vous tentez de créer un autre volume de swap, créez un volume ZFS d'une taille spécifique et activez le swap sur le périphérique. Ajoutez ensuite une entrée pour le nouveau périphérique de swap dans le fichier `/etc/vfstab`. Par exemple :

```
# zfs create -V 2G rpool/swap2
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile                   dev  swaplo blocks   free
/dev/zvol/dsk/rpool/swap  256,1      16 2097136 2097136
```

```
/dev/zvol/dsk/rpool/swap2 256,5      16 4194288 4194288
```

N'effectuez pas de swap vers un fichier dans un système de fichiers ZFS. La configuration de fichier swap ZFS n'est pas prise en charge.

Pour plus d'informations sur l'ajustement de la taille des volumes de swap et de vidage, reportez-vous à la section “Ajustement de la taille de vos périphériques de swap et de vidage ZFS” à la page 122.

Utilisation d'un volume ZFS en tant qu'unité logique de stockage iSCSI

Le logiciel COMSTAR (Common Multiprotocol SCSI Target) permet de convertir n'importe quel hôte Oracle Solaris en périphérique cible SCSI accessible à des hôtes initiateurs via un réseau de stockage. Vous pouvez créer et configurer un volume ZFS en vue de le partager en tant qu'unité logique de stockage (LUN) iSCSI.

Commencez par installer le package COMSTAR.

```
# pkg install group/feature/storage-server
```

Créez ensuite un volume ZFS qui sera utilisé en tant que cible iSCSI, puis créez le LUN basé sur un périphérique en mode bloc SCSI. Par exemple :

```
# zfs create -V 2g tank/volumes/v2
# sbdadm create-lu /dev/zvol/rdsk/tank/volumes/v2
Created the following LU:

GUID                    DATA SIZE           SOURCE
--------------------------------  -------------------  ----------------
600144f000144f1dafaa4c0faff20001  2147483648           /dev/zvol/rdsk/tank/volumes/v2
# sbdadm list-lu
Found 1 LU(s)

GUID                    DATA SIZE           SOURCE
--------------------------------  -------------------  ----------------
600144f000144f1dafaa4c0faff20001  2147483648           /dev/zvol/rdsk/tank/volumes/v2
```

Vous pouvez exposer les vues du LUN à tous les clients ou à des clients sélectionnés. Identifiez le GUID du LUN, puis partagez la vue du LUN. Dans l'exemple suivant, la vue du LUN est partagée avec tous les clients.

```
# stmfadm list-lu
LU Name: 600144F000144F1DAFAA4C0FAFF20001
# stmfadm add-view 600144F000144F1DAFAA4C0FAFF20001
# stmfadm list-view -l 600144F000144F1DAFAA4C0FAFF20001
View Entry: 0
Host group   : All
Target group : All
LUN          : 0
```

L'étape suivante consiste à créer les iSCSI cibles. Pour plus d'informations sur la création de cibles iSCI, reportez-vous au Chapitre 8, “ Configuration des périphériques de stockage avec COMSTAR ” du manuel “ Gestion des périphériques dans Oracle Solaris 11.2 ”.

Un volume ZFS en tant que cible iSCSI est géré comme n'importe quel autre jeu de données ZFS, à l'exception du fait que vous ne pouvez pas renommer l'ensemble de données, annuler une capture d'écran de volume, ou de l'exportation du pool pendant que les volumes ZFS sont partagés en tant que iSCSI LUN. Des messages similaires au message suivant s'afficheront :

```
# zfs rename tank/volumes/v2 tank/volumes/v1
cannot rename 'tank/volumes/v2': dataset is busy
# zpool export tank
cannot export 'tank': pool is busy
```

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 :

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 permet également la délégation de jeux de données à une zone non globale, ce qui permet de contrôler parfaitement le jeu de données et tous ses enfants à l'administrateur de zone. 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 de jeux de données non ajoutés à la zone, y compris ceux qui dépassent les quotas de niveau supérieur du jeu de données délégué.

Tenez compte des points suivants lorsque vous utilisez ZFS sur un système sur lequel des zones Oracle Solaris sont installées :

- La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`.
- 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 “ Création et utilisation d’Oracle Solaris Zones ”.

Ajout de systèmes de fichiers ZFS à une zone non globale

Vous pouvez ajouter un système de fichiers ZFS en tant que système de fichiers générique lorsqu'il s'agit simplement de partager de l'espace avec la zone globale. La propriété `mountpoint` d'un système de fichiers ZFS ajouté à une zone non globale doit être définie sur `legacy`. Par exemple, si le système de fichiers `tank/zone/zion` doit être ajouté à une zone non globale, définissez comme suit la propriété `mountpoint` dans la zone globale :

```
# zfs set mountpoint=legacy tank/zone/zion
```

Vous pouvez ajouter un système de fichiers ZFS à une zone non globale à l'aide de la sous-commande`add fs` de la commande `zonecfg`.

Dans l'exemple suivant, un système de fichiers ZFS est ajouté à une zone non globale par un administrateur global de la zone globale :

```
# zonecfg -z zion
zonecfg:zion> add fs
zonecfg:zion:fs> set type=zfs
zonecfg:zion:fs> set special=tank/zone/zion
zonecfg:zion:fs> set dir=/opt/data
zonecfg:zion:fs> end
```

Cette syntaxe permet d'ajouter le système de fichiers ZFS `tank/zone/zion` à la zone `zion` déjà configurée et montée sur `/opt/data`. La propriété `mountpoint` du système de fichiers doit être définie sur `legacy` et le système de fichiers ne peut pas être déjà monté à un autre emplacement. L'administrateur de zone peut créer et détruire des fichiers au sein du système de fichiers. Le système de fichiers ne peut pas être remonté à un autre emplacement, tout comme l'administrateur ne peut pas modifier les propriétés suivantes du système de fichiers : `atime`, `readonly`, `compression`, etc.

L'administrateur de zone globale est chargé de la configuration et du contrôle des propriétés du système de fichiers.

Pour plus d'informations sur la commande`zonecfg` et les types ressources de configuration avec `zonecfg`, reportez-vous à “ Création et utilisation d’Oracle Solaris Zones ”.

Délégation de jeux de données à une zone non globale

Si l'objectif principal est de déléguer l'administration du stockage d'une zone, 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> set alias=tank
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. Dans la zone `zion`, ce système de fichiers n'est pas accessible en tant que `tank/zone/zion`, mais en tant que *virtual pool* nommé `tank`. L'alias du système de fichiers délégué fournit à la zone en tant que pool virtuel une vue du pool d'origine. La propriété d'alias indique le nom du pool virtuel. Si aucun alias n'est précisé, un alias par défaut correspondant au dernier composant du nom du système de fichiers est utilisé. Si aucun alias n'avait été indiqué, l'alias par défaut aurait été `zion` dans l'exemple ci-dessus.

Dans les jeux de données délégués, l'administrateur des zones peut définir les propriétés des systèmes de fichiers et créer des systèmes de fichiers descendants. En outre, l'administrateur des zones peut créer des instantanés ainsi que des clones, et contrôler la totalité de la hiérarchie du système de fichiers. Si des volumes ZFS sont créés au sein de systèmes de fichier délégués, ils risquent d'entrer en conflit avec les volumes ZFS ajoutés en tant que ressources de périphériques. Pour plus d'informations, consultez la section suivante.

Ajout de volumes ZFS à une zone non globale

Vous pouvez ajouter ou créer un volume ZFS dans une zone non globale ou vous pouvez ajouter l'accès aux données d'un volume dans une zone non globale de l'une des manières suivantes :

- Dans une zone non globale, un administrateur de zone possédant des privilèges peut créer un volume ZFS en tant que descendant d'un système de fichiers précédemment délégué. Par exemple :

  ```
  # zfs create -V 2g tank/zone/zion/vol1
  ```

  La syntaxe ci-dessus signifie que l'administrateur des zones peut gérer les propriétés du volume et des données dans la zone non globale.
- Dans une zone globale, utilisez la sous-commande `zonecfg add dataset` et indiquez un volume ZFS à ajouter à une zone non globale. Par exemple :

  ```
  # zonecfg -z zion
  zonecfg:zion> add dataset
  zonecfg:zion:dataset> set name=tank/volumes/vol1
  zonecfg:zion:dataset> end
  ```

  La syntaxe ci-dessus signifie que l'administrateur des zones peut gérer les propriétés du volume et des données dans la zone non globale.
- Dans une zone globale, utilisez la sous-commande `zonecfg add device` et indiquez un volume ZFS dont les données sont accessibles dans une zone non globale. Par exemple :

  ```
  # zonecfg -z zion
  zonecfg:zion> add device
  zonecfg:zion:device> set match=/dev/zvol/dsk/tank/volumes/vol2
  zonecfg:zion:device> end
  ```

  La syntaxe précédente signifie que seules les données du volume peuvent être consultées dans la zone non globale.

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 ayant l'alias `zion` par défaut, chaque jeu de données possède les propriétés suivantes.

Jeu de données	visible	Accessible en écriture	Propriétés immuables
`tank`	Non	-	-
`tank/home`	Non	-	-
`tank/data`	Non	-	-
`tank/data/zion`	Yes	Yes	`zoned`, `quota`, `reservation`
`tank/data/zion/home`	Yes	Yes	`zoned`

Notez que tous les parents de `tank/zone/zion` sont invisibles et que tous ses descendants sont accessibles en écriture. L'administrateur de zone ne peut pas modifier la propriété `zoned` car cela entraînerait un risque de sécurité, comme 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`

Lorsqu'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. La propriété `zoned` permet à ZFS d'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.

Copie de zones vers d'autres systèmes

Si vous devez migrer une ou plusieurs zones vers un autre système, pensez à utiliser les commandes `zfs send` et `zfs receive`. Selon le scénario, il peut être préférable d'utiliser des flux de réplication ou des flux récursifs.

Les exemples de cette section décrivent la copie de données de zone d'un système à un autre. Des étapes supplémentaires sont nécessaires pour transférer la configuration de chaque zone et rattacher chaque zone au nouveau système. Pour plus d'informations, reportez-vous à “ Création et utilisation d’Oracle Solaris Zones ”.

Si toutes les zones d'un système doivent migrer vers un autre système, envisagez d'utiliser un flux de réplication car il permet de conserver les instantanés et les clones. Les instantanés et les clones sont largement utilisés par les commandes `pkg update`, `beadm create` et `zoneadm clone`.

Dans l'exemple suivant, les zones de `sysA` sont installées dans le système de fichiers `rpool/zones` et doivent être copiées dans le système de fichiers `tank/zones` sur `sys`. Les commandes suivantes créent un instantané et copient les données vers `sysb` à l'aide d'un flux de réplication :

```
sysA# zfs snapshot -r rpool/zones@send-to-sysB
sysA# zfs send -R rpool/zones@send-to-sysB | ssh sysB zfs receive -d tank
```

Dans l'exemple ci-dessous, l'une des zones est copiée de `sysC` vers `sysD`. Supposons que la commande `ssh` ne soit pas disponible mais qu'une instance de serveur NFS le soit. Les commandes suivantes peuvent être utilisées pour générer un flux `zfs send` récursif sans se soucier de savoir si la zone est le clone d'une autre zone ou non.

```
sysC# zfs snapshot -r rpool/zones/zone1@send-to-nfs
sysC# zfs send -rc rpool/zones/zone1@send-to-nfs > /net/nfssrv/export/scratch/zone1.zfs
sysD# zfs create tank/zones
sysD# zfs receive -d tank/zones < /net/nfssrv/export/scratch/zone1.zfs
```

Utilisation d'un pool ZFS avec un emplacement de root de remplacement

Lors de sa création, un pool est intrinsèquement lié au système hôte. Le système hôte gère les informations du pool. Cela lui permet de détecter l'indisponibilité de ce dernier, le cas échéant. Même si elles sont utiles dans des conditions normales d'utilisation, ces informations peuvent causer des interférences lors de l'initialisation à partir d'autres médias ou lors de la création d'un pool sur un média amovible. La fonction de pool *root de remplacement* de ZFS permet de résoudre ce problème. Un pool root de remplacement n'y a pas de lieu subsistent après plusieurs réinitialisation du système et tous les points de montage sont modifiés de sorte à être relatifs à la root du pool.

Création d'un pool ZFS avec un emplacement de root de remplacement

La création d'un pool root de remplacement s'effectue le plus souvent en vue d'une utilisation avec un média amovible. Dans ces circonstances, les utilisateurs souhaitent employer un système de fichiers unique et le monter à l'emplacement de leur choix dans le système cible. Lors de la création d'un pool à l'aide de l'option `zpool create`-R, le point de montage du système de fichiers root est automatiquement défini sur `/`, l'équivalent de la valeur de root de remplacement lui-même.

Dans l'exemple suivant, un pool nommé `morpheus` est créé à l'aide `/mnt` en tant que chemin root de remplacement :

```
# zpool create -R /mnt morpheus c0t0d0
# zfs list morpheus
NAME                  USED  AVAIL  REFER  MOUNTPOINT
morpheus             32.5K  33.5G     8K  /mnt
```

Notez le système de fichiers `morpheus` dont le point de montage est l'emplacement root de remplacement du pool, `/mnt`. Le point de montage stocké sur le disque est `/`, et le chemin complet de `/mnt` n'est interprété que dans le contexte initial de création de pool. Ce système de fichiers peut ensuite être exporté ou importé sous un emplacement root de remplacement de remplacement arbitraire d'un autre système en respectant la syntaxe de *valeur de root secondaire*-R.

```
# zpool export morpheus
# zpool import morpheus
cannot mount '/': directory is not empty
# zpool export morpheus
# zpool import -R /mnt morpheus
# zfs list morpheus
```

```
NAME                  USED  AVAIL  REFER  MOUNTPOINT
morpheus             32.5K  33.5G     8K  /mnt
```

Importation d'un pool avec emplacement root de remplacement

L'importation de pool s'effectue également à l'aide d'un emplacement root de remplacement. Cette fonction permet de récupérer les données, le cas échéant, lorsque les points de montage ne doivent pas être interprétés dans le contexte due root actuel, mais sous un répertoire temporaire où pourront s'effectuer les réparations. Vous pouvez également utiliser cette fonction lors 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 point de montage de root de remplacement : Cet exemple part du principe que `morpheus` a été précédemment exporté.

```
# zpool import -R /a pool
# zpool list morpheus
NAME   SIZE   ALLOC  FREE    CAP  HEALTH  ALTROOT
pool  44.8G    78K  44.7G     0%  ONLINE  /a
# zfs list pool
NAME   USED  AVAIL  REFER  MOUNTPOINT
pool  73.5K  44.1G    21K  /a/pool
```

Importation d'un pool avec un nom temporaire

En plus de l'importation d'un pool dans un autre emplacement root, vous pouvez importer un pool avec un nom temporaire. Dans certains de stockage partagé ni de récupérer les données, le cas échéant, cette fonction permet deux nom de pools présentant les mêmes simultanément persistant importé. L'un de ces pools doivent être importés sous un nom temporaire.

Dans l'exemple suivant, le pool `rpool` est importé à un emplacement root de remplacement sous un nom temporaire. Des conflits de nom car le pool permanent à un pool qui est déjà importé, il doit être importé par pool ou en indiquant les périphériques ID.

```
# zpool import
pool: rpool
id: 16760479674052375628
state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

rpool     ONLINE
c8d1s0  ONLINE
# zpool import -R /a -t altrpool 16760479674052375628
```

```
# zpool list
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
altrpool   97G  22.4G   74G  23%  1.00x  ONLINE  /a
rpool     465G  75.1G  390G  16%  1.00x  ONLINE  -
```

Un pool peuvent également être créé sous un nom temporaire à l'aide de l'option `zpool create -t`.

♦ ♦ ♦ CHAPITRE 10

Dépannage d'Oracle Solaris ZFS et récupération de pool

Ce chapitre décrit les méthodes d'identification et de résolution des pannes de ZFS. Des informations relatives à la prévention des pannes sont également fournies.

Ce chapitre contient les sections suivantes :

Pour plus d'informations sur la récupération complète de pool root, reportez-vous à“ Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ”.

Identification des problèmes ZFS

En tant que système de fichiers et gestionnaire de volumes combinés, ZFS peut rencontrer différentes pannes. Ce chapitre indique comment diagnostiquer les pannes matérielles générales et comment résoudre les problèmes liés au système de fichiers et au périphérique de pool. Vous pouvez rencontrer les types de problèmes suivants :

- **Problèmes matériels généraux** : les problèmes matériels peuvent affecter les performances de votre pool et la disponibilité des données de votre pool. Eliminez les problèmes matériels généraux, tels que la mémoire et les composants défectueux, avant de déterminer les problèmes à un niveau supérieur, tel qu'au niveau de vos pools et systèmes de fichiers.
- **Problèmes de pool de stockage ZFS**
  - “Identification des problèmes avec les pools de stockage ZFS” à la page 286
  - “Résolution des problèmes des périphériques de stockage ZFS” à la page 292

- **Les données sont endommagées :**“Résolution des problèmes de données dans un pool de stockage ZFS” à la page 307
- **La configuration est endommagée** :“Réparation d'une configuration ZFS endommagée” à la page 318
- **Le système ne s'initialise pas :**“Réparation d'un système impossible à réinitialiser” à la page 319

Notez que les trois types d'erreurs peuvent se produire dans un même pool. Une procédure de réparation complète implique de trouver et de corriger une erreur, de passer à la suivante et ainsi de suite.

Résolution des problèmes matériels généraux

Consultez les sections suivantes pour déterminer si des problèmes de pool ou une indisponibilité de système de fichiers sont liés à un problème matériel, tel qu'une carte système, une mémoire, un périphérique, un HBA défectueux ou une erreur de configuration.

Par exemple, un disque défaillant ou défectueux situé sur un pool ZFS occupé peut considérablement dégrader les performances globales du système.

Si vous commencez par diagnostiquer et identifier les problèmes matériels, qui peuvent être plus faciles à détecter, et par effectuer une vérification matérielle, vous pouvez ensuite diagnostiquer les problèmes liés au pool et aux systèmes de fichiers comme décrit dans le reste de ce chapitre. Si vos configurations matérielles, de pool et de système de fichiers sont en bon état, pensez à diagnostiquer les problèmes d'application, qui sont généralement plus complexes à corriger et qui ne sont pas décrits dans ce guide.

Identification des pannes du matériel et des périphériques

Le gestionnaire des pannes Solaris recherche les problèmes logiciels, matériels et spécifiques aux périphériques en identifiant les informations de télémétrie des erreurs qui indiquent un symptôme spécifique dans un journal des erreurs puis signalent le diagnostic de panne réelle lorsque le symptôme d'erreur entraîne une panne réelle.

La commande suivante identifie toute panne logicielle ou matérielle.

```
# fmadm faulty
```

Exécutez régulièrement la commande ci-dessus pour identifier les services ou les périphériques défectueux.

Exécutez la commande suivante régulièrement pour identifier les erreurs liées au matériel ou aux périphériques.

```
# fmdump -eV | more
```

Les messages d'erreur de ce fichier journal qui décrivent les problèmes `vdev.open_failed`, `checksum` ou `io_failure` requièrent votre attention ou ils pourraient se transformer en pannes réelles affichées avec la commande de défaut `fmadm`.

Si cette procédure indique qu'un périphérique est défectueux, c'est le bon moment pour vous assurer de la disponibilité d'un périphérique de remplacement.

Vous pouvez également suivre les autres erreurs de périphérique à l'aide de commande `iostat`. Utilisez la syntaxe suivante afin d'identifier un résumé des statistiques d'erreurs.

```
# iostat -en
---- errors ---
s/w h/w trn tot device
0   0   0   0 c0t5000C500335F95E3d0
0   0   0   0 c0t5000C500335FC3E7d0
0   0   0   0 c0t5000C500335BA8C3d0
0  12   0  12 c2t0d0
0   0   0   0 c0t5000C500335E106Bd0
0   0   0   0 c0t50015179594B6F11d0
0   0   0   0 c0t5000C500335DC60Fd0
0   0   0   0 c0t5000C500335F907Fd0
0   0   0   0 c0t5000C500335BD117d0
```

Dans la sortie ci-dessus, les erreurs sont signalées sur le disque interne `c2t0d0`. Utilisez la syntaxe suivante pour afficher des erreurs de périphérique plus détaillées.

Erreurs de transport de la résolution persistant ou non persistante

Persistant des erreurs de transport SCSI ou jusqu'à ce qu'il restaure qui font référence à la valeur du champ Nombre de tentatives vers le bas '', peut être à l'origine de problèmes, par un disque défectueux révision du microprogramme, un mauvais matériel câble, ou d'une connexion. Certaines erreurs de transport transitoire peut être résolu par la mise à niveau de votre HBA ou les microprogrammes des périphériques. Si des erreurs de transport et persistent après que tous les périphériques microprogrammes sont mis à jour, puis consulter sont considérées comme étant opérationnelles câble ou un autre pour une connexion entre les composants matériels défectueux.

Rapport système de messages d'erreur ZFS

Outre le suivi permanent des erreur au sein du pool, ZFS affiche également des messages `syslog` lorsque des événements intéressants se produisent. Les scénarios suivants donnent lieu à des événements de notification, procédez comme suit :

- **Transition d'état de périphérique** : si l'état d'un périphérique devient `FAULTED`, ZFS consigne un message indiquant que la tolérance de pannes du pool risque d'être compromise. Un message similaire est envoyé si le périphérique est mis en ligne ultérieurement, restaurant la maintenance du pool.
- **Altération de données** : en cas de détection d'altération de données, ZFS consigne un message indiquant où et quand s'est produit la détection. Ce message n'est consigné que lors de la première détection. Les accès ultérieurs ne génèrent pas de message.
- **Défaillances de pool et de périphérique** : en cas de défaillance d'un pool ou d'un périphérique, le démon du gestionnaire de pannes rapporte ces erreurs par le biais de messages `syslog` et de la commande `fmdump`.

Si ZFS détecte un erreur de périphérique et la corrige automatiquement, aucune notification n'est générée. De telles erreurs ne constituent pas une défaillance de redondance de pool ou de l'intégrité des données. En outre, de telles erreurs sont typiquement dues à un problème de pilote accompagné de son propre jeu de messages d'erreur.

Identification des problèmes avec les pools de stockage ZFS

Les sections suivantes décrivent l'identification et la résolution des problèmes dans les systèmes de fichiers ZFS ou les pools de stockage :

- “Recherche de problèmes éventuels dans un pool de stockage ZFS” à la page 288
- “Collecte des informations sur l'état du pool de stockage ZFS” à la page 288
- “Rapport système de messages d'erreur ZFS” à la page 285

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.
- Un pool de stockage ZFS détruit accidentellement peut être récupéré à l'aide de la commande `zpool import -D`, mais il est important que le pool soit récupérée rapidement de manière à ce que les périphériques ne soient pas réutilisés et écrasés. Pour obtenir des informations supplémentaires, reportez-vous à la section “Récupération de pools de stockage ZFS détruits” à la page 100. Récupérer similaire fonctionnalité n'est disponible pour aucun des fichiers ou des données ZFS. Toujours ayez une bonne des sauvegardes.

La commande `zpool status` permet de résoudre la plupart des problèmes au niveau de ZFS. Cette commande analyse les différentes erreurs système et identifie les problèmes les plus

sévères. En outre, elle propose des actions à effectuer et un lien vers un article de connaissances pour de plus amples informations. Notez que cette commande n'identifie qu'un seul problème dans le pool, même si plusieurs problèmes existent. Par exemple, les erreurs d'altération de données sont généralement provoquées par la panne d'un périphérique, mais le remplacement d'un périphérique défaillant peut ne pas résoudre tous les problèmes d'altération de données.

En outre, un moteur de diagnostic ZFS diagnostique et signale les défaillances au niveau du pool et des périphériques. Les erreurs liées aux sommes de contrôle, aux E/S, aux périphériques et aux pools font également l'objet d'un rapport lorsqu'elles sont liées à ces défaillances. Les défaillances ZFS telles que rapportées par `fmd` s'affichent sur la console ainsi que les dans le fichier de messages système. Dans la plupart des cas, le message `fmd`vous invite à la commande `zpool status`pour obtenir des instructions supplémentaires de récupération.

Le processus de récupération est comme décrit ci-après :

- Le cas échéant, la commande `zpool history` permet d'identifier les commandes ZFS ayant précédé le scénario d'erreur. Par exemple :

  ```
  # zpool history tank
  History for 'tank':
  2012-11-12.13:01:31 zpool create tank mirror c0t1d0 c0t2d0 c0t3d0
  2012-11-12.13:28:10 zfs create tank/eric
  2012-11-12.13:37:48 zfs set checksum=off tank/eric
  ```

  Dans cette sortie, notez que les sommes de contrôle sont désactivées pour le système de fichiers `tank/eric`. Cette configuration est déconseillée.
- Identifiez les erreurs à l'aide des messages `fmd` affichés sur la console système ou dans le fichier `/var/adm/messages`.
- Obtenez des instructions de réparation supplémentaires grâce à la commande `zpool status -x`.
- Réparez les pannes. Pour ce faire, suivez les étapes ci-après :
  - Remplacez le périphérique indisponible ou manquant et mettez-le en ligne.
  - Restaurez la configuration défaillante ou les données endommagées 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 plus d'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 78.

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

Les informations d'état du pool de stockage ZFS s'affichent à l'aide de la commande `zpool status`. Par exemple :

```
# zpool status pond
pool: pond
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
Run 'zpool status -v' to see device specific details.
scan: scrub repaired 0 in 0h0m with 0 errors on Wed Jun 20 13:16:09 2012
config:

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

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

Cette sortie est décrite dans la section suivante.

Informations globales d'état des pools

Cette section dans 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.
`Etat`	Indique l'état de maintenance actuel du pool. Ces informations concernent uniquement la capacité de pool à fournir le niveau de réplication requis.
`état`	Décrit les problèmes du pool. Ce champ est absent si aucune erreur n'est détectée.
`action`	Action recommandée pour la réparation des erreurs. Ce champ est absent si aucune erreur n'est détectée.
`voir`	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.
`nettoyage`	Identifie l'état actuel d'une opération de nettoyage. Ce champ peut indiquer la date et l'heure du dernier nettoyage, un nettoyage en cours ou l'absence de demande de nettoyage.
`erreurs`	Identifie les erreurs de données ou l'absence d'erreurs de données connues.

Informations sur la configuration des pools de stockage ZFS

Le champ `config` de la sortie `zpool status` décrit la configuration des périphériques inclus dans le pool, ainsi que leur état et toute erreur générée à partir des périphériques. L'état peut être l'un des suivants : `ONLINE`, `FAULTED`, `DEGRADED` ou `SUSPENDED`. Si l'état n'est pas `ONLINE`, la tolérance de pannes du pool a été compromise.

La deuxième section de la sortie de configuration affiche des statistiques d'erreurs. Ces erreurs se divisent en trois catégories :

- `READ` : erreurs d'E/S qui se sont produites lors de l'envoi d'une demande de lecture
- `WRITE` : erreurs d'E/S qui se sont produites lors de l'envoi d'une demande d'écriture
- `CKSUM` : erreurs de somme de contrôle signifiant que le périphérique a renvoyé des données 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 une altération de données telle qu'interprétée par les applications. Si la configuration du périphérique est redondante, les périphériques peuvent présenter des erreurs impossibles à corriger, même si aucune erreur ne s'affiche au niveau du périphérique RAID-Z ou du miroir. Dans ce cas, ZFS a récupéré les données correctes et a réussi à réparer les données endommagées à partir des répliques existantes.

Pour plus d'informations sur l'interprétation de ces erreurs, reportez-vous à la section “Détermination du type de panne de périphérique” à la page 296.

Enfin, les informations auxiliaires supplémentaire sont affichées dans la dernière colonne de la sortie de `zpool status`. Ces informations s'étendent dans le champ `state` et facilitent le diagnostic des pannes. Si l'état d'un périphérique est `UNAVAIL`, ce champ indique si le périphérique est inaccessible ou si les données du périphérique sont endommagées. Si le périphérique est en cours de réargenture, ce champ affiche la progression du processus.

Pour plus d'informations sur la surveillance de la progression de la réargenture, reportez-vous à la section “Affichage de l'état de réargenture” à la page 305.

Etat du nettoyage des pools de stockage ZFS

La section sur le nettoyage de la sortie `zpool status` décrit l'état actuel de toute opération de nettoyage explicite. Ces informations sont distinctes de la détection d'erreurs dans le système, mais il est possible de les utiliser pour déterminer l'exactitude du rapport d'erreurs d'altération de données. Si le dernier nettoyage s'est récemment terminé, toute altération de données existante aura probablement déjà été détectée.

Les messages d'état du nettoyage `zpool status` suivants sont fournis :

- Rapport de progression du nettoyage. Par exemple :

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

- Message de fin du nettoyage. Par exemple :

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

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

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

Les messages de fin de nettoyage subsistent après plusieurs réinitialisations du système.

Pour plus d'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 310.

Erreurs d'altération de données ZFS

La commande `zpool status` indique également si des erreurs connues sont associées au pool. La détection de ces erreurs a pu s'effectuer lors du nettoyage des données ou lors des opérations normales. Le système de fichiers ZFS gère un journal persistant de toutes les erreurs de données associées à un pool. Ce journal tourne à chaque fois qu'un nettoyage complet du système est terminé.

Les erreurs d'altération de données constituent toujours des erreurs fatales. Elles indiquent une erreur d'E/S dans au moins une application, en raison de la présence de données endommagées au sein du pool. Les erreurs de périphérique dans un pool redondant n'entraînent pas d'altération de données et ne sont pas enregistrées en tant que partie de ce journal. Par défaut, seul le nombre d'erreurs trouvées s'affiche. Vous pouvez obtenir la liste complète des erreurs et de leurs spécificités à l'aide de l'option `zpool status -v.` Par exemple :

```
# zpool status -v tank
pool: tank
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption.  Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://support.oracle.com/msg/ZFS-8000-8A
scan: scrub repaired 0 in 0h0m with 2 errors on Fri Jun 29 16:58:58 2012
config:

NAME           STATE     READ WRITE CKSUM
tank           ONLINE       2     0     0
c8t0d0         ONLINE       0     0     0
c8t1d0         ONLINE       2     0     0

errors: Permanent errors have been detected in the following files:

/tank/file.1
```

La commande `fmd` affiche également un message similaire dans la console système et le fichier `/var/adm/messages.` La commande `fmdump` permet également de réaliser le suivi de ces messages.

Pour plus d'informations sur l'interprétation d'erreurs d'altération de données, reportez-vous à la section “Identification du type d'altération de données” à la page 314.

Résolution des problèmes des périphériques de stockage ZFS

Consultez les sections suivantes pour résoudre un problème de périphérique manquant, supprimé ou défectueux.

Résolution d'un périphérique manquant ou supprimé

Si l'ouverture d'un périphérique est impossible, ce dernier s'affiche dans l'état `UNAVAIL` dans la sortie de `zpool status`. Cet état indique que ZFS n'a pas pu ouvrir le périphérique lors du premier accès au pool ou que le périphérique est devenu indisponible par la suite. Si le périphérique rend un périphérique de niveau supérieur indisponible, l'intégralité du pool devient inaccessible. Dans le cas contraire, la tolérance de pannes du pool risque d'être compromise. Quel que soit le cas, le périphérique doit simplement être reconnecté au système pour fonctionner à nouveau normalement. Si vous avez besoin de remplacer un périphérique qui est `UNAVAIL` parce qu'il est défectueux, reportez-vous à la section “Remplacement d'un périphérique dans un pool de stockage ZFS” à la page 299.

Si l'état d'un périphérique est `UNAVAIL` dans un pool root ou dans un pool root mis en miroir, consultez les références suivantes :

- **La mise en miroir du disque de pool root a échoué** : “Initialisation à partir d'un disque alternatif d'un pool root ZFS mis en miroir” à la page 125
- **Remplacement d'un disque dans un pool root**
  - “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/EFI (GPT))” à la page 117
  - “Remplacement d'un disque dans un pool root ZFS (SPARC ou x86/EFI (GPT))” à la page 117
- **Récupération sur sinistre de pool root complet** : “ Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris 11.2 ”.

Par exemple, après une panne de périphérique, `fmd` peut afficher un message similaire au suivant :

```
SUNW-MSG-ID: ZFS-8000-QJ, TYPE: Fault, VER: 1, SEVERITY: Minor
EVENT-TIME: Wed Jun 20 13:09:55 MDT 2012
PLATFORM: ORCL,SPARC-T3-4, CSN: 1120BDRCCD, HOSTNAME: tardis
```

```
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: e13312e0-be0a-439b-d7d3-cddaefe717b0
DESC: Outstanding dtls on ZFS device 'id1,sd@n5000c500335dc60f/a' in pool 'pond'.
AUTO-RESPONSE: No automated response will occur.
IMPACT: None at this time.
REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this event.
Run 'zpool status -lx' for more information. Please refer to the associated
reference document at http://support.oracle.com/msg/ZFS-8000-QJ for the latest
service procedures and policies regarding this diagnosis.
```

Pour afficher des informations détaillées sur le problème du périphérique et sa résolution, utilisez la commande `zpool status-v`. Par exemple :

```
# zpool status -v
pool: pond
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
scan: scrub repaired 0 in 0h0m with 0 errors on Wed Jun 20 13:16:09 2012
config:

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

device details:

c0t5000C500335DC60Fd0    UNAVAIL          cannot open
status: ZFS detected errors on this device.
The device was missing.
see: http://support.oracle.com/msg/ZFS-8000-LR for recovery
```

Dans cette sortie, vous pouvez voir que le périphérique `c0t5000C500335DC60Fd0` ne fonctionne pas. Si vous estimez que le périphérique est défectueux, remplacez-le.

Si nécessaire, exécutez ensuite la commande `zpool online` pour mettre le périphérique remplacé en ligne. Par exemple :

```
# zpool online pond c0t5000C500335DC60Fd0
```

Signalez à FMA que le périphérique a été remplacé si la sortie de `fmadm faulty` identifie l'erreur de périphérique. Par exemple :

```
# fmadm faulty
```

```
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID         SEVERITY
--------------- ------------------------------------  -------------- ---------
Jun 20 13:15:41 3745f745-371c-c2d3-d940-93acbb881bd8  ZFS-8000-LR    Major

Problem Status    : solved
Diag Engine       : zfs-diagnosis / 1.0
System
Manufacturer  : unknown
Name          : ORCL,SPARC-T3-4
Part_Number   : unknown
Serial_Number : 1120BDRCCD
Host_ID       : 84a02d28

----------------------------------------
Suspect 1 of 1 :
Fault class : fault.fs.zfs.open_failed
Certainty   : 100%
Affects     : zfs://pool=86124fa573cad84e/
  vdev=25d36cd46e0a7f49/pool_name=pond/
  vdev_name=id1,sd@n5000c500335dc60f/a
Status      : faulted and taken out of service

FRU
Name             : "zfs://pool=86124fa573cad84e/
vdev=25d36cd46e0a7f49/pool_name=pond/
vdev_name=id1,sd@n5000c500335dc60f/a"
Status        : faulty

Description : ZFS device 'id1,sd@n5000c500335dc60f/a'
in pool 'pond' failed to open.

Response    : An attempt will be made to activate a hot spare if available.

Impact      : Fault tolerance of the pool may be compromised.

Action      : Use 'fmadm faulty' to provide a more detailed view of this event.
Run 'zpool status -lx' for more information. Please refer to the
associated reference document at
http://support.oracle.com/msg/ZFS-8000-LR for the latest service
procedures and policies regarding this diagnosis.
```

Extrayez la chaîne dans la section `Affects:` de la sortie `fmadm faulty` et incluez-la dans la commande suivante pour informer FMA que le périphérique est remplacé :

```
# fmadm repaired zfs://pool=86124fa573cad84e/ \
   vdev=25d36cd46e0a7f49/pool_name=pond/ \
   vdev_name=id1,sd@n5000c500335dc60f/a
fmadm: recorded repair to of zfs://pool=86124fa573cad84e/
   vdev=25d36cd46e0a7f49/pool_name=pond/vdev_
name=id1,sd@n5000c500335dc60f/a
```

Confirmez ensuite que le pool dont le périphérique a été remplacé fonctionne correctement. Par exemple :

```
# zpool status -x tank
pool 'tank' is healthy
```

Résolution d'un périphérique supprimé

Si un périphérique est entièrement supprimé du système, ZFS s'assure que le périphérique ne peut pas être ouvert et il le place dans l'état `REMOVED`. En fonction du niveau de réplication des données du pool, ce retrait peut résulter ou non en une indisponibilité de la totalité du pool. Le pool reste accessible en cas de suppression d'un périphérique mis en miroir ou RAID-Z. Un pool peut renvoyer l'état `UNAVAIL`, cela signifie qu'aucune donnée n'est accessible jusqu'à ce que le périphérique soit reconnecté selon les conditions suivantes :

Périphérique de pool de stockage si redondant, reportez-vous à la est accidentellement supprimé et réinséré, vous pouvez, il vous suffit d'effacer l'erreur de périphérique dans la plupart des cas. Par exemple :

```
# zpool clear tank c1t1d0
```

Reconnexion physique d'un périphérique

La reconnexion d'un périphérique dépend du périphérique en question. S'il s'agit d'un disque connecté au réseau, la connectivité au réseau doit être restaurée. S'il s'agit d'un périphérique USB ou autre média amovible, il doit être reconnecté au système. S'il s'agit d'un disque local, un contrôleur est peut-être tombé en panne, rendant le périphérique invisible au système. Dans ce cas, il faut remplacer le contrôleur pour que les disques soient à nouveau disponibles. D'autres problèmes existent et dépendent du type de matériel et de sa configuration. Si un disque tombe en panne et n'est plus visible pour le système, le périphérique doit être traité comme un périphérique endommagé. Suivez les procédures décrites dans la section “Remplacement ou réparation d'un périphérique endommagé ” à la page 296.

Un pool peut être `SUSPENDED` si la connectivité du périphérique est problématique. Un pool `SUSPENDED` reste en état wait jusqu'à ce que le problème du périphérique soit résolu. Par exemple :

```
# zpool status cybermen
pool: cybermen
state: SUSPENDED
status: One or more devices are unavailable in response to IO failures.
The pool is suspended.
action: Make sure the affected devices are connected, then run 'zpool clear' or
'fmadm repaired'.
Run 'zpool status -v' to see device specific details.
see: http://support.oracle.com/msg/ZFS-8000-HC
scan: none requested
config:
```

```
NAME        STATE     READ WRITE CKSUM
cybermen    UNAVAIL      0    16     0
c8t3d0      UNAVAIL      0     0     0
c8t1d0      UNAVAIL      0     0     0
```

Une fois la connectivité de périphérique restaurée, effacez les erreurs de pool ou de périphérique.

```
# zpool clear cybermen
# fmadm repaired zfs://pool=name/vdev=guid
```

Notification relative à la disponibilité de périphériques dans ZFS

Une fois le périphérique reconnecté au système, sa disponibilité peut être détectée automatiquement ou non dans ZFS. Si le pool était précédemment `UNAVAIL` ou `SUSPENDED`, ou si le système a été réinitialisé en tant que partie de la procédure `attach`, ZFS rebalaye automatiquement tous les périphériques lors de la tentative d'ouverture du pool. Si le pool était endommagé et que le périphérique a été remplacé alors que le système était en cours d'exécution, vous devez indiquer à ZFS que le périphérique est dorénavant disponible et qu'il est prêt à être rouvert à l'aide de la commande `zpool online`. Par exemple :

```
# zpool online tank c0t1d0
```

Pour plus d'informations sur la remise en ligne de périphériques, reportez-vous à la section “Mise en ligne d'un périphérique” à la page 66.

Remplacement ou réparation d'un périphérique endommagé

Cette section explique comment déterminer les types de panne de périphériques, effacer les erreurs transitoires et remplacer un périphérique.

Détermination du type de panne de périphérique

L'expression *périphérique endommagé* est plutôt vague et 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 altérations de données potentielles dans des systèmes de grande taille ou de longue durée.

- **Lectures ou écritures mal dirigées** : les bogues de microprogrammes ou les pannes de matériel peuvent entraîner un référencement incorrect de l'emplacement du disque par des lectures ou écritures de blocs entiers. Ces erreurs sont généralement transitoires, mais un grand nombre d'entre elles peut indiquer un disque défectueux.
- **Erreur d'administrateur** : les administrateurs peuvent écraser par erreur des parties du disque avec des données erronées (la copie de `/dev/zero` sur des parties du disque, par exemple) qui entraînent l'endommagement permanent 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 endommagements aléatoires ou des pannes. Ces erreurs sont typiquement permanentes.
- **Périphérique mis hors ligne** : si un périphérique est hors ligne, il est considéré comme ayant été mis hors ligne par l'administrateur, parce qu'il était défectueux. L'administrateur qui a mis ce dispositif hors ligne peut déterminer si cette hypothèse est exacte.

Il est parfois difficile de déterminer la nature exacte de la panne du dispositif. La première étape consiste à examiner le décompte d'erreurs dans la sortie de `zpool status`. Par exemple :

```
# zpool status -v tank
pool: tank
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption.  Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://support.oracle.com/msg/ZFS-8000-8A
config:

NAME           STATE     READ WRITE CKSUM
tank           ONLINE       2     0     0
c8t0d0         ONLINE       0     0     0
c8t0d0         ONLINE       2     0     0

errors: Permanent errors have been detected in the following files:

/tank/file.1
```

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 ZFS endommagées” à la page 313.

Erreurs de périphérique Non persistant ou Persistant de compensation

Si les erreurs de périphérique sont considérées comme étant transitoires, dans la mesure où il est peu probable qu'elles affectent la maintenance du périphérique, elles peuvent être effacées en toute sécurité pour indiquer qu'aucune erreur fatale ne s'est produite. Pour effacer les compteurs d'erreurs pour les périphériques mis en miroir ou RAID-Z, utilisez la commande `zpool clear`. Par exemple :

```
# zpool clear tank c1t1d0
```

Cette syntaxe efface toutes les erreurs du périphérique et tout décompte d'erreurs de données associées au périphérique.

Pour effacer toutes les erreurs associées aux périphériques virtuels du pool et tout décompte d'erreurs de données associées au pool, respectez la syntaxe suivante :

```
# zpool clear tank
```

Pour plus d'informations sur la suppression des erreurs de pool, reportez-vous à la section “Effacement des erreurs de périphérique de pool de stockage” à la page 67.

Des erreurs de périphérique transitoires sont probablement supprimées à l'aide de la commande `zpool clear`. Si un périphérique a n'a pas été soumis, reportez-vous à la section suivante sur le remplacement d'un périphérique. Si un périphérique redondant a été accidentellement écrasé ou était UNAVAIL pendant une longue période, il est possible que l'erreur doit être résolu à l'aide de la commande, `fmadm repaired` selon les instructions du message de sortie.`zpool status`. Par exemple :

```
# zpool status -v pond
pool: pond
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a
degraded state.
action: Determine if the device needs to be replaced, and clear the errors
using 'zpool clear' or 'fmadm repaired', or replace the device
with 'zpool replace'.
scan: scrub repaired 0 in 0h0m with 0 errors on Wed Jun 20 15:38:08 2012
config:

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

device details:

c0t5000C500335F907Fd0    UNAVAIL          cannot open
status: ZFS detected errors on this device.
The device was missing.
see: http://support.oracle.com/msg/ZFS-8000-LR for recovery


errors: No known data errors
```

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 300
- “Périphériques impossibles à remplacer” à la page 300
- “Remplacement d'un périphérique dans un pool de stockage ZFS” à la page 301

- “Affichage de l'état de réargenture” à la page 305

Détermination de la possibilité de remplacement du périphérique

Si le périphérique à remplacer fait partie d'une configuration redondante, il doit exister suffisamment de répliques pour permettre la récupération des données correctes. Si deux disques d'un miroir à quatre directions sont `UNAVAIL`, chaque disque peut être remplacé car des répliques saines sont disponibles. Cependant, si deux disques dans un périphérique virtuel RAID-Z à quatre directions (`raidz1`) sont `UNAVAIL`, aucun disque ne peut être remplacé en l'absence de répliques suffisantes permettant de récupérer les données. Si le périphérique est endommagé mais en ligne, il peut être remplacé tant que l'état du pool n'est pas `UNAVAIL`. Toutefois, toute donnée endommagée sur le périphérique est copiée sur le nouveau périphérique, à moins que copies des données non endommagées soit déjà suffisant.

Dans la configuration suivante, le disque `c1t1d0` peut être remplacé et toutes les données du pool sont copiées à partir de la réplique saine, `c1t0d0`.

```
    mirror            DEGRADED
c1t0d0             ONLINE
c1t1d0             UNAVAIL
```

Le disque `c1t0d0` peut également être remplacé, mais un autorétablissement des données est impossible, car il n'existe aucune réplique correcte.

Dans la configuration suivante, aucun des disques `UNAVAIL` ne peut être remplacé. Les disques `ONLINE` ne peuvent pas l'être non plus, car le pool lui-même est `UNAVAIL`.

```
    raidz1              UNAVAIL
c1t0d0              ONLINE
c2t0d0              UNAVAIL
c3t0d0              UNAVAIL
c4t0d0              ONLINE
```

Dans la configuration suivante, chacun des disques de niveau supérieur peut être remplacé. Cependant, les données incorrectes seront également copiées dans le nouveau disque, le cas échéant.

```
c1t0d0         ONLINE
c1t1d0         ONLINE
```

Si les deux disques sont `UNAVAIL`, tout remplacement est impossible car le pool lui-même est `UNAVAIL`.

Périphériques impossibles à remplacer

Si la perte d'un périphérique rend un pool `UNAVAIL` ou si le périphérique contient trop d'erreurs de données dans une configuration non redondante, le remplacement du périphérique en toute

sécurité est impossible. En l'absence de redondance suffisante, il n'existe pas de données correctes avec lesquelles réparer le périphérique défectueux. Dans ce cas, la seule option est de détruire le pool, recréer la configuration et restaurer les données à partir d'une copie de sauvegarde.

Pour plus d'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. Si vous remplacez le périphérique endommagé par un autre périphérique différent, utilisez la syntaxe ci-dessous :

```
# zpool replace tank c1t1d0 c2t0d0
```

Cette commande lance la migration de données vers le nouveau périphérique, soit à partir du périphérique endommagé, soit à partir d'autres périphériques du pool s'il s'agit d'une configuration redondante. Une fois l'exécution de la commande terminée, le périphérique endommagé est séparé de la configuration. Il peut dorénavant être retiré du système. Si vous avez déjà retiré le périphérique et que vous l'avez remplacé par un autre dans le même emplacement, utilisez la forme "périphérique unique" de la commande. Par exemple :

```
# zpool replace tank c1t1d0
```

Cette commande formate adéquatement un disque non formaté et resynchronise ensuite les données à partir du reste de la configuration.

Pour plus d'informations sur la commande `zpool replace` reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 67.

EXEMPLE 10-1 Remplacement d'un disque SATA dans un pool de stockage ZFS

L'exemple suivant illustre le remplacement d'un périphérique (`c1t3d0`) du pool de stockage mis en miroir `tank` sur un système équipé de périphériques SATA. Pour remplacer le disque `c1t3d0` par un nouveau au même emplacement (`c1t3d0`), annulez la configuration du disque avant de procéder au remplacement. Si le disque qui doit être remplacé n'est pas un disque SATA, reportez-vous à la section “Remplacement de périphériques dans un pool de stockage” à la page 67.

Voici les principales étapes à suivre :

- Déconnectez le disque (`c1t3d0`) à remplacer. Vous ne pouvez pas annuler la configuration d'un disque SATA utilisé.
- Utilisez la commande `cfgadm` pour identifier le disque SATA (`c1t3d0`) dont la configuration doit être annulée et annulez-la. Dans cette configuration en miroir, le pool est endommagé et le disque est hors ligne, mais le pool reste disponible.

- Remplacez le disque (`c1t3d0`). Vérifiez que la DEL bleue `Ready to Remove`, indiquant que le périphérique est prêt à être retiré, est allumée avant de retirer le lecteur `UNAVAIL`, si disponible.
- Reconfigurez le disque SATA (`c1t3d0`).
- Mettez le nouveau disque (`c1t3d0`) en ligne.
- Exécutez la commande `zpool replace` pour remplacer le disque (`c1t3d0`).

Remarque - Si vous avez précédemment défini la propriété de pool `autoreplace` sur `on`, tout nouveau périphérique détecté au même emplacement physique qu'un périphérique appartenant précédemment au pool est automatiquement formaté et remplacé sans recourir à la commande `zpool replace`. Cette fonction n'est pas prise en charge sur tous les types de matériel.

- Si un disque défectueux est automatiquement remplacé par un disque hot spare, vous devrez peut-être déconnecter le disque hot spare une fois le disque défectueux remplacé. Par exemple, si `c2t4d0` reste actif comme disque hot spare actif une fois le disque défectueux remplacé, déconnectez-le.

  ```
  # zpool detach tank c2t4d0
  ```

- Si FMA signale le périphérique défaillant, effacez la panne de périphérique.

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

L'exemple suivant explique étape par étape comment remplacer un disque dans un pool de stockage ZFS.

```
# zpool offline tank c1t3d0
# cfgadm | grep c1t3d0
sata1/3::dsk/c1t3d0            disk         connected    configured   ok
# cfgadm -c unconfigure sata1/3
Unconfigure the device at: /devices/pci@0,0/pci1022,7458@2/pci11ab,11ab@1:3
This operation will suspend activity on the SATA device
Continue (yes/no)? yes
# cfgadm | grep sata1/3
sata1/3                        disk         connected    unconfigured ok
<Physically replace the failed disk c1t3d0>
# cfgadm -c configure sata1/3
# cfgadm | grep sata1/3
sata1/3::dsk/c1t3d0            disk         connected    configured   ok
# zpool online tank c1t3d0
# zpool replace tank c1t3d0
# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:17:32 2010
config:

NAME      STATE     READ WRITE CKSUM
```

```
tank      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
c0t1d0    ONLINE       0     0     0
c1t1d0    ONLINE       0     0     0
mirror-1  ONLINE       0     0     0
c0t2d0    ONLINE       0     0     0
c1t2d0    ONLINE       0     0     0
mirror-2  ONLINE       0     0     0
c0t3d0    ONLINE       0     0     0
c1t3d0    ONLINE       0     0     0

errors: No known data errors
```

Notez que la commande `zpool output` affiche parfois l'ancien disque et le nouveau sous l'en-tête de *remplacement*. Par exemple :

```
replacing     DEGRADED     0     0    0
c1t3d0s0/o  FAULTED      0     0    0
c1t3d0      ONLINE       0     0    0
```

Ce texte signifie que la procédure de remplacement et la réargenture du nouveau disque sont en cours.

Pour remplacer un disque (`c1t3d0`) par un autre disque (`c4t3d0`), il suffit d'exécuter la commande `zpool replace`. Par exemple :

```
# zpool replace tank c1t3d0 c4t3d0
# zpool status
pool: tank
state: DEGRADED
scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:

NAME       STATE     READ WRITE CKSUM
tank       DEGRADED     0     0     0
mirror-0   ONLINE       0     0     0
c0t1d0     ONLINE       0     0     0
c1t1d0     ONLINE       0     0     0
mirror-1   ONLINE       0     0     0
c0t2d0     ONLINE       0     0     0
c1t2d0     ONLINE       0     0     0
mirror-2   DEGRADED     0     0     0
c0t3d0     ONLINE       0     0     0
replacing  DEGRADED     0     0     0
c1t3d0     OFFLINE      0     0     0
c4t3d0     ONLINE       0     0     0

errors: No known data errors
```

La commande `zpool status` doit parfois être exécutée plusieurs fois jusqu'à la fin du remplacement du disque.

```
# zpool status tank
```

```
pool: tank
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Tue Feb  2 13:35:41 2010
config:

NAME      STATE     READ WRITE CKSUM
tank      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
c0t1d0    ONLINE       0     0     0
c1t1d0    ONLINE       0     0     0
mirror-1  ONLINE       0     0     0
c0t2d0    ONLINE       0     0     0
c1t2d0    ONLINE       0     0     0
mirror-2  ONLINE       0     0     0
c0t3d0    ONLINE       0     0     0
c4t3d0    ONLINE       0     0     0
```

EXEMPLE 10-2 Remplacement d'un périphérique de journalisation défaillant

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

L'exemple suivant montre comment récupérer les données d'un périphérique de journalisation défaillant (`c0t5d0`) dans le pool de stockage (`pool`). Voici les principales étapes à suivre :

- Passez en revue `zpool status` les messages de diagnostic de sortie FMA `-x`, tel que décrits dans la rubrique *échec de lecture du journal ZFS (Doc ID 1021625.1)* dans https://support.oracle.com/.
- Remplacez physiquement le périphérique de journalisation défaillant.
- Mettez le nouveau périphérique de journalisation en ligne.
- Effacez la condition d'erreur du pool.
- Effacez l'erreur FMA.

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

```
# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
Waiting for adminstrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:

NAME      STATE     READ WRITE CKSUM
pool      FAULTED      0     0     0 bad intent log
```

```
mirror-0  ONLINE       0     0     0
c0t1d0    ONLINE       0     0     0
c0t4d0    ONLINE       0     0     0
logs      FAULTED      0     0     0 bad intent log
c0t5d0    UNAVAIL      0     0     0 cannot open
<Physically replace the failed log device>
# zpool online pool c0t5d0
# zpool clear pool
# fmadm faulty
# fmadm repair zfs://pool=name/vdev=guid
```

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

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

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

  ```
  # zpool clear pool
  ```

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

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

Affichage de l'état de réargenture

Le processus de remplacement d'un périphérique peut prendre beaucoup de temps, selon la taille du périphérique et la quantité de données dans le pool. Le processus de déplacement de données d'un périphérique à un autre s'appelle la *réargenture*, et vous pouvez la surveiller à l'aide de la commande `zpool status`.

Les messages d'état de la réargenture `zpool status` suivants sont fournis :

- Rapport de progression de la réargenture. Par exemple :

  ```
  scan: resilver in progress since Mon Jun  7 09:17:27 2010
  13.3G scanned
  13.3G resilvered at 18.5M/s, 82.34% done, 0h2m to go
  ```
- Message de fin de la réargenture. Par exemple :

```
    resilvered 16.2G in 0h16m with 0 errors on Mon Jun  7 09:34:21 2010
```

Les messages de fin de réargenture subsistent après plusieurs réinitialisations du système.

Les systèmes de fichiers traditionnels effectuent la réargenture de données au niveau du bloc. Dans la mesure où ZFS élimine la séparation en couches artificielles du gestionnaire de volume, il peut effectuer la réargenture de façon bien plus puissante et contrôlée. Les deux avantages de cette fonction sont comme suite :

- ZFS n'effectue la réargenture que de la quantité minimale de données requises. Dans le cas d'une brève interruption de service (par rapport à un remplacement complet d'un périphérique), vous pouvez effectuer la réargenture du disque en quelques minutes ou quelques secondes. Lors du remplacement d'un disque entier, la durée du processus de réargenture est proportionnelle à la quantité de données utilisées dans le disque. Le remplacement d'un disque de 500 Go ne dure que quelques secondes si le pool ne contient que quelques giga-octets d'espace utilisé.
- En cas de mise hors-tension ou de réinitialisation du système, le processus de réargenture reprend exactement là où il s'est arrêté, sans requérir une intervention manuelle.

La commande `zpool status` permet de visualiser le processus de réargenture. Par exemple :

```
# zpool status tank
pool: tank
state: 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.
scan: resilver in progress since Mon Jun  7 10:49:20 2010
54.6M scanned54.5M resilvered at 5.46M/s, 24.64% done, 0h0m to go

config:

NAME         STATE     READ WRITE CKSUM
tank         ONLINE       0     0     0
mirror-0     ONLINE       0     0     0
replacing-0  ONLINE       0     0     0
c1t0d0       ONLINE       0     0     0
c2t0d0       ONLINE       0     0     0  (resilvering)
c1t1d0       ONLINE       0     0     0
```

Dans cet exemple, le disque `c1t0d0` est remplacé par `c2t0d0`. Cet événement est observé dans la sortie d'état par la présence du périphérique virtuel `replacing` de la configuration. Ce périphérique n'est pas réel et ne permet pas de créer un pool. L'objectif de ce périphérique consiste uniquement à afficher le processus de réargenture et à identifier le périphérique en cours de remplacement.

Notez que tout pool en cours de réargenture se voit attribuer l'état `ONLINE` ou `DEGRADED`, car il ne peut pas fournir le niveau souhaité de redondance tant que le processus n'est pas terminé. La réargenture s'effectue aussi rapidement que possible, mais les E/S sont toujours programmées

avec une priorité inférieure à celle des E/S requises par l'utilisateur afin de minimiser l'impact sur le système. Une fois la réargenture terminée, la nouvelle configuration complète s'applique, remplaçant l'ancienne configuration. Par exemple :

```
# zpool status tank
pool: tank
state: ONLINE
scrub: resilver completed after 0h1m with 0 errors on Tue Feb  2 13:54:30 2010
config:

NAME      STATE     READ WRITE CKSUM
tank      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
c2t0d0    ONLINE       0     0     0  377M resilvered
c1t1d0    ONLINE       0     0     0

errors: No known data errors
```

L'état du pool est à nouveau `ONLINE` et le disque défectueux d'origine (`c1t0d0`) a été supprimé de la configuration.

Modification de périphériques de pool

N'essayez pas de modifier les périphériques sous un pool actif pool.

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

Un autre problème est que si vous tentez de modifier les dispositifs associés à une mise en commun, puis vous utilisez la commande `zpool status` en tant qu'utilisateur autre que root, les noms de périphériques précédents peuvent être affichés.

Résolution des problèmes de données dans un pool de stockage ZFS

Parmi les exemples de problèmes de données sont les suivantes :

- L'espace des systèmes de fichiers pool est manquant ou

- Erreurs d'E/S transitoires causées par un disque ou un contrôleur défaillant
- Altération de données sur disque causée par les rayons cosmiques
- Bogues de pilotes entraînant des transferts de données vers ou à partir d'un emplacement erroné
- Ecrasement accidentel de parties du périphérique physique par un utilisateur

Certaines erreurs sont transitoires, par exemple une erreur d'E/S aléatoire alors que le contrôleur rencontre des problèmes. Dans d'autres cas, les dommages sont permanents, par exemple lors de l'endommagement sur disque. En outre, même si les dommages sont permanents, cela ne signifie pas que l'erreur est susceptible de se reproduire. Par exemple, si un utilisateur écrase une partie d'un disque par accident, aucune panne matérielle ne s'est produite et il est inutile de remplacer le périphérique. L'identification du problème exact dans un périphérique n'est pas une tâche aisée. Elle est abordée plus en détail dans une section ultérieure.

Résolution des problèmes d'espace ZFS

Consultez les sections suivantes si vous n'êtes pas sûr de la manière dont ZFS signale le système de fichiers et la comptabilisation d'espace du pool. Consultez également la section “Comptabilisation de l'espace disque ZFS” à la page 20.

Compte-rendu d'espace de système de fichiers ZFS

Les commandes `zpool list` et `zfs list` sont plus appropriées que les commandes précédentes `df` et `du` pour déterminer l'espace disponible des pools et des systèmes de fichiers. Les anciennes commandes ne permettent pas de distinguer facilement l'espace des pools de l'espace des systèmes de fichiers. D'autre part, elles ne tiennent pas compte de l'espace utilisé par les systèmes de fichiers descendants ou les instantanés.

Par exemple, le pool root ci-après (`rpool`) utilise 5,46 Go et dispose de 68,5 Go d'espace libre.

```
# zpool list rpool
NAME   SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool   74G  5.46G  68.5G   7%  1.00x  ONLINE  -
```

Si vous comparez la comptabilisation d'espace de pool à la comptabilisation d'espace de système de fichiers en consultant la colonne `USED` de vos systèmes de fichiers individuels, vous pouvez voir que l'espace de pool qui est signalé dans `ALLOC` est globalement présent dans le total `USED` du système de fichiers. Par exemple :

```
# zfs list -r rpool
NAME                     USED  AVAIL  REFER  MOUNTPOINT
rpool                   5.41G  67.4G  74.5K  /rpool
rpool/ROOT              3.37G  67.4G    31K  legacy
rpool/ROOT/solaris      3.37G  67.4G  3.07G  /
```

```
rpool/ROOT/solaris/var    302M  67.4G   214M  /var
rpool/dump               1.01G  67.5G  1000M  -
rpool/export             97.5K  67.4G    32K  /rpool/export
rpool/export/home        65.5K  67.4G    32K  /rpool/export/home
rpool/export/home/admin  33.5K  67.4G  33.5K  /rpool/export/home/admin
rpool/swap               1.03G  67.5G  1.00G  -
```

Compte-rendu sur l'espace des pools de stockage ZFS

La valeur de taille SIZE calculée par la commande `zpool list` indique généralement la quantité d'espace disque physique dans le pool, mais elle varie selon le niveau de redondance de celui-ci. Voir les exemples ci-dessous. La commande `zfs list` liste l'espace disponible pour des systèmes de fichiers, c'est-à-dire l'espace disque moins l'espace utilisé par les métadonnées de gestion de la redondance des pools ZFS, le cas échéant.

Les configurations de jeux de données ZFS suivantes sont suivies par la commande `zfs list`, mais ils ne sont pas suivies en tant qu'espace alloué dans la sortie `zpool list` :

- Quota d'un système de fichiers ZFS
- Réservation de système de fichiers ZFS
- La taille du volume logique ZFS

Les éléments suivants d'impact décrivent comment les différentes configurations de pool agissent sur les sorties `zpool list` et `zfs list` :

- **Pool de stockage non redondant** : un pool est créé avec un seul disque de 136 Go ; la commande `zpool list` signale une valeur de taille `SIZE` et une valeur d'espace libre initiale `FREE` de 136 Go. L'espace disponible initial `AVAIL` indiqué par la commande `zfs list` est de 134 Go, en raison d'une petite quantité de métadonnées de gestion de pool. Par exemple :

  ```
  # zpool create tank c0t6d0
  # zpool list tank
  NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
  tank   136G  95.5K   136G     0%  1.00x  ONLINE  -
  # zfs list tank
  NAME   USED  AVAIL  REFER  MOUNTPOINT
  tank    72K   134G    21K  /tank
  ```

- **Pool de stockage mis en miroir** : quand un pool est créé avec deux disques de 136 Go , la commande `zpool list` indique une valeur de taille `SIZE` de 136 Go et une valeur d'espace libre initiale `FREE` de 136 Go. Ce compte-rendu est appelé valeur d'espace *minorée*. L'espace disponible initial `AVAIL` indiqué par la commande `zfs list` est de 134 Go, en raison d'une petite quantité de métadonnées de gestion de pool. Par exemple :

  ```
  # zpool create tank mirror c0t6d0 c0t7d0
  # zpool list tank
  ```

```
NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
tank   136G  95.5K   136G     0%  1.00x  ONLINE  -
# zfs list tank
NAME   USED  AVAIL  REFER  MOUNTPOINT
tank    72K   134G    21K  /tank
```

- **Pool de stockage RAID-Z** : quand un pool `raidz2` est créé avec trois disques de 136 Go ; la commande `zpool list` signale une valeur de taille `SIZE` de 408 Go et une valeur d'espace libre initiale `FREE` de 408 Go. Ce compte-rendu est appelé valeur d'espace disque *majorée*, qui inclut l'espace nécessaire à la gestion de la redondance, par exemple les informations de parité. L'espace disponible initial `AVAIL` indiqué par la commande `zfs list` est de 133 GO, en raison de la gestion de la redondance du pool. La différence d'espace entre les sorties `zpool list` et `zfs list` pour un pool RAID-Z provient du fait que `zpool list` indique l'espace de pool majoré.

  ```
  # zpool create tank raidz2 c0t6d0 c0t7d0 c0t8d0
  # zpool list tank
  NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
  tank   408G   286K   408G     0%  1.00x  ONLINE  -
  # zfs list tank
  NAME   USED  AVAIL  REFER  MOUNTPOINT
  tank  73.2K   133G  20.9K  /tank
  ```

  Pour plus d'informations sur l''effet des changements de `recordsize` sur la comptabilisation d'espace, reportez-vous à la section “Comptabilisation de l'espace disque ZFS” à la page 20.

- **Espace du système de fichiers monté NFS** : ni le compte `zpool list` ou le compte `zfs list` ne sont est garant pour l'espace pour système de fichiers monté NFS. Toutefois, les fichiers de données peuvent se local masqués par une système de fichiers monté NFS. S'il vous manque d'espace de système de fichiers, assurez-vous que vous ne disposez pas des fichiers de données masqués par un système de fichiers NFS.

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. Le seul moyen dont dispose incohérence des données sur le disque, dans une configuration ZFS s'effectue par l'intermédiaire de 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'un endommagement du 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 manière la plus simple de contrôler l'intégrité des données est d'initier un nettoyage explicite de toutes les données du pool. Cette opération traverse toutes les données dans le pool une fois et vérifie que tous les blocs sont lisibles. Le nettoyage va aussi vite que le permettent les périphériques, mais la priorité de toute E/S reste inférieure à celle de toute opération normale.

Cette opération peut affecter les performances, bien que les données du pool restent utilisables et leur réactivité quasiment la même lors du nettoyage. La commande `zpool scrub`permet de lancer un nettoyage explicite. Par exemple :

```
# zpool scrub tank
```

La commande `zpool status` ne permet pas d'afficher l'état de l'opération de nettoyage actuelle. Par exemple :

```
# zpool status -v tank
pool: tank
state: ONLINE
scan: scrub in progress since Mon Jun  7 12:07:52 2010
201M scanned out of 222M at 9.55M/s, 0h0m to go
0 repaired, 90.44% done
config:

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

errors: No known data errors
```

Une seule opération de nettoyage actif par pool peut se produire à la fois.

L'option `-s` permet d'interrompre une opération de nettoyage en cours. Par exemple :

```
# zpool scrub -s tank
```

Dans la plupart des cas, une opération de nettoyage pour assurer l'intégrité des données doit être menée à son terme. Vous pouvez cependant interrompre une telle opération si les performances du système sont affectées.

Un nettoyage de routine garantit des E/S continues pour l'ensemble des disques du système. Cet opération a cependant pour effet secondaire d'empêcher la gestion de l'alimentation de placer des disques inactifs en mode basse consommation. Si le système réalise en général des E/S en permanence, ou si la consommation n'est pas une préoccupation, ce problème peut être ignoré. Si le système est en grande partie inactif et que vous souhaitez conserver l'alimentaiton des disques, vous pouvez envisager d'utiliser un nettoyage explicite cron(1M) programmé plutôt qu'un nettoyage d'arrière-plan. Gérer intégralement ce scrubs est toujours l'organisation de fabrication, bien qu'elle de uniquement les données générera une grande quantité de nettoyage d'E / S est terminée, jusqu'à ce que le pour que les disques en suivant la procédure normale peut être la gestion de l'alimentation. Le côté négatif (en plus d'augmenter les E/S) est qu'il y aura de longues périodes pendant lesquelles aucun nettoyage ne sera effectué, ce qui augmentera le risque potentiel d'altération au cours de ces pérodes.

Pour plus d'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 78.

Nettoyage et réargenture de données ZFS

Lors du remplacement d'un périphérique, une opération de réargenture est amorcée pour déplacer les données des copies correctes vers le nouveau périphérique. Cette action est une forme de nettoyage de disque. Par conséquent, une seule action de ce type peut être effectuée à un moment donné dans le pool. Lorsqu'une opération de nettoyage est en cours, toute opération de réargenture suspend le nettoyage ; le nettoyage reprend une fois que la réargenture est terminée.

Pour plus d'informations sur la réargenture, reportez-vous à la section “Affichage de l'état de réargenture” à la page 305.

Réparation de données ZFS endommagées

L'altération de données se produit lorsqu'une ou plusieurs erreurs de périphériques (indiquant un ou plusieurs périphériques manquants ou endommagés) affectent un périphérique virtuel de niveau supérieur. Par exemple, la moitié d'un miroir peut subir des milliers d'erreurs sans jamais causer d'altération des données. Si une erreur se produit sur l'autre côté du miroir au même emplacement, les données sont endommagées.

L'altération des 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 endommagé est supprimé, le nettoyage de disque suivant reconnaît que l'altération n'est plus présente et supprime toute trace de l'erreur dans le système.

Les sections suivantes décrivent comment identifier le type d'altération de données et comment réparer les données le cas échéant.

- “Identification du type d'altération de données” à la page 314
- “Réparation d'un fichier ou répertoire endommagé” à la page 315
- “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'autorétablissement pour minimiser le risque d'altération de données. Cependant, l'altération de données peut se produire si le pool n'est pas redondant, si une altération s'est produite alors que le pool était endommagé ou si une série d'événements improbables a endommagé plusieurs copies d'un élément de données. Quelle que soit la source, le résultat est le même : les données sont

endommagées et par conséquent inaccessibles. Les actions à effectuer dépendent du type de données endommagées et de leurs valeurs relatives. Deux types de données peuvent être endommagés :

- 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 endommagées, le pool entier ou des parties de la hiérarchie du jeu de données sont indisponibles.
- Données d'objet : dans ce cas, l'altération se produit au sein d'un fichier ou périphérique spécifique. Ce problème peut rendre une partie du fichier ou répertoire inaccessible ou endommager l'objet.

Les données sont vérifiées lors des opérations normales et lors du nettoyage. Pour plus d'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 310.

Identification du type d'altération de données

Par défaut, la commande `zpool status` indique qu'une altération s'est produite, mais n'indique pas à quel endroit. Par exemple :

```
# zpool status tank
pool: tank
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption.  Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://support.oracle.com/msg/ZFS-8000-8A
config:

NAME                      STATE     READ WRITE CKSUM
tank                      ONLINE       4     0     0
c0t5000C500335E106Bd0     ONLINE       0     0     0
c0t5000C500335FC3E7d0     ONLINE       4     0     0


errors: 2 data errors, use '-v' for a list
```

Toute erreur indique seulement qu'une erreur s'est produite à un moment donné. Il est possible que certaines erreurs ne soient plus présentes dans le système. Dans le cadre d'une utilisation normale, elles le sont. Certaines interruptions de service temporaires peuvent entraîner une altération de données qui est automatiquement réparée une fois l'interruption de service terminée. Un nettoyage complet du pool examine chaque bloc actif dans le pool. Ainsi, le journal d'erreur est réinitialisé à la fin de chaque nettoyage. Si vous déterminez que les erreurs ne sont plus présentes et ne souhaitez pas attendre la fin du nettoyage, la commande `zpool online` permet de réinitialiser toutes les erreurs du pool.

Si l'altération de données se produit dans des métadonnées au niveau du pool, la sortie est légèrement différente. Par exemple :

```
# zpool status -v morpheus
pool: morpheus
id: 13289416187275223932
state: UNAVAIL
status: The pool metadata is corrupted.
action: The pool cannot be imported due to damaged devices or data.
see: http://support.oracle.com/msg/ZFS-8000-72
config:

morpheus  FAULTED   corrupted data
c1t10d0   ONLINE
```

Dans le cas d'une altération au niveau du pool, ce dernier se voit attribuer l'état `FAULTED`, car le pool ne peut pas fournir le niveau de redondance requis.

Réparation d'un fichier ou répertoire endommagé

En cas d'altération d'un fichier ou d'un répertoire, le système peut tout de même continuer à fonctionner, selon le type d'altération. Tout dommage est irréversible, à moins que des copies correctes des données n'existent sur le système. Si les données sont importantes, vous devez restaurer les données affectées à partir d'une sauvegarde. Quand bien même, vous devriez pouvoir réparer les données endommagées sans restaurer la totalité du pool.

En cas de dommages au sein d'un bloc de données de fichiers, le fichier peut être supprimé en toute sécurité. L'erreur est alors effacée du système. Utilisez la commande `zpool status -v` pour afficher la liste des noms de fichier contenant des erreurs persistantes. Par exemple :

```
# zpool status tank -v
pool: tank
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption.  Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://support.oracle.com/msg/ZFS-8000-8A
config:

NAME                     STATE     READ WRITE CKSUM
tank                     ONLINE       4     0     0
c0t5000C500335E106Bd0    ONLINE       0     0     0
c0t5000C500335FC3E7d0    ONLINE       4     0     0

errors: Permanent errors have been detected in the following files:
/tank/file.1
/tank/file.2
```

La liste des noms de fichiers comportant des erreurs persistantes peut être décrite comme suit :

- Si le chemin complet du fichier est trouvé et si le jeu de données est monté, le chemin complet du fichier s'affiche. Par exemple :

  ```
  /monkey/a.txt
  ```

- Si chemin complet du fichier est trouvé mais que le jeu de données n'est pas monté, le nom du jeu de données non précédé d'un slash (/) s'affiche, suivi du chemin du fichier au sein du jeu de données. Par exemple :

  ```
  monkey/ghost/e.txt
  ```

- Si le nombre d'objet vers un chemin de fichiers ne peut pas être converti, soit en raison d'une erreur soit parce qu'aucun chemin de fichiers réel n'est associé à l'objet, tel que c'est le cas pour `dnode_t`, alors le nom du jeu de données s'affiche, suivi du numéro de l'objet. Par exemple :

  ```
  monkey/dnode:<0x0>
  ```

- En cas d'altération d'un MOS (Meta-Object Set, jeu de méta-objet), la balise spéciale `<metadata>` s'affiche, suivie du numéro de l'objet.

Vous pouvez tenter de résoudre des altérations de données mineures à l'aide du nettoyage de pool et de la suppression des erreurs de pool dans plusieurs itérations. Effacez et nettoyez si la première itération n'est pas paramétré pour résoudre le again.For des fichiers corrompus exemple :, exécutez-les

```
# zpool scrub tank
# zpool clear tank
```

Si l'altération 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 données endommagées avec plusieurs références de blocs

Si un système de fichiers endommagé contient des données endommagées avec plusieurs références de blocs tels que les instantanés, la commande `zpool status -v` ne peut pas afficher les chemins de **toutes** les données endommagées. La génération de rapports `zpool status` actuelle de données endommagées est limitée par le niveau d'altération de métadonnées et par la réutilisation d'un bloc après l'exécution de la commande `zpool status`. Les blocs dédupliqués rendent la génération de rapports sur les données endommagées encore plus complexe.

Si vous avez des données endommagées, et que la commande `zpool status- v` signale que les données d'instantanées sont affectées, pensez à exécuter la commande suivante afin d'identifier les autres chemins endommagés :

Réparation de dommages présents dans l'ensemble du pool de stockage ZFS

Si des dommages sont présents dans les métadonnées du pool et que cela empêche l'ouverture ou l'importation du pool, vous pouvez utiliser les options suivantes :

- Tentez de récupérer le pool à l'aide de la commande `zpool clear-F` ou `zpool import-F`. Ces commandes tentent d'annuler (roll back) les dernières transactions restantes du pool pour qu'elles reviennent à un fonctionnement normal. Vous pouvez utiliser la commande `zpool status` pour vérifier le pool endommagé et les mesures de récupération recommandées. Par exemple :

```
# zpool status
pool: tpool
state: UNAVAIL
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 Fri Jun 29 17:22:49 2012
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://support.oracle.com/msg/ZFS-8000-72
scrub: none requested
config:

NAME      STATE     READ WRITE CKSUM
tpool     UNAVAIL      0     0     1  corrupted data
c1t1d0    ONLINE       0     0     2
c1t3d0    ONLINE       0     0     4
```

Le processus de récupération comme décrit dans la sortie ci-dessus consiste à utiliser la commande suivante :

```
# zpool clear -F tpool
```

Si vous tentez d'importer un pool de stockage endommagé, des messages semblables aux messages suivants s'affichent :

```
# zpool import tpool
cannot import 'tpool': I/O error
Recovery is possible, but will result in some data loss.
Returning the pool to its state as of Fri Jun 29 17:22:49 2012
should correct the problem.  Approximately 5 seconds of data
must be discarded, irreversibly.  Recovery can be attempted
by executing 'zpool import -F tpool'.  A scrub of the pool
```

```
is strongly recommended after recovery.
```

Le processus de récupération comme décrit dans la sortie ci-dessus consiste à utiliser la commande suivante :

```
# zpool import -F tpool
Pool tpool returned to its state as of Fri Jun 29 17:22:49 2012.
Discarded approximately 5 seconds of transactions
```

Si le pool endommagé se trouve dans le fichier `zpool.cache`, le problème est détecté lors de l'initialisation du système. Le pool endommagé est consigné dans la commande `zpool status`. Si le pool ne se trouve pas dans le fichier `zpool.cache`, il n'est pas importé ou ouvert et des messages indiquant que le pool est endommagé s'affichent lorsque vous tentez de l'importer.

- Vous pouvez importer un pool endommagé en mode lecture seule. Cette méthode permet d'importer le pool, ce qui vous permet d'accéder aux données. Par exemple :

  ```
  # zpool import -o readonly=on tpool
  ```

  Pour plus d'informations sur l'importation d'un pool en lecture seule, reportez-vous à la section “Importation d'un pool en mode lecture seule” à la page 99.
- Vous pouvez importer un pool avec un périphérique de journalisation manquant à l'aide de la commande `zpool import-m`. Pour plus d'informations, reportez-vous à la section “Importation d'un pool avec un périphérique de journalisation manquant” à la page 97.
- Si le pool ne peut pas être récupéré par le biais de l'une des méthodes de récupération de pool, vous devez restaurer le pool et l'ensemble de ses données à partir d'une copie de sauvegarde. Le mécanisme utilisé varie énormément selon la configuration du pool et la stratégie de sauvegarde. Tout d'abord, enregistrez la configuration telle qu'elle s'affiche dans la commande `zpool status` pour pouvoir la recréer après la destruction du pool. Ensuite, détruisez le pool à l'aide de la commande `zpool destroy-f`.

  Gardez une description de 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 que vous utilisez.

Réparation d'une configuration ZFS endommagée

ZFS gère un cache de pools actifs et les configurations correspondantes dans le système de fichiers root. Si ce fichier de cache est altéré ou n'est plus synchronisé avec les informations de configuration stockées dans le disque, l'ouverture du pool n'est plus possible. Le système de fichiers ZFS tente d'éviter cette situation, même si des altérations arbitraires peuvent

toujours survenir en raison des caractéristiques du système de fichiers sous-jacent et du stockage. En général, cette situation est due à la disparition d'un pool du système alors qu'il devrait être disponible. Parfois, elle correspond à une configuration partielle, dans laquelle il manque un nombre inconnu de périphériques virtuels de niveau supérieur. Dans les deux cas, la configuration peut être récupérée en exportant le pool (s'il est visible à tous) et en le réimportant.

Pour plus d'informations sur l'importation et l'exportation de pools, reportez-vous à la section “Migration de pools de stockage ZFS” à la page 92.

Réparation d'un système impossible à réinitialiser

ZFS a été conçu pour être robuste et stable malgré les erreurs. Cependant, les bogues de logiciels ou certains problèmes inattendus peuvent entraîner la panique du système lors de l'accès à un pool. Dans le cadre du processus d'initialisation, chaque pool doit être ouvert. En raison de ces défaillances, le système effectue des réinitialisations en boucle. Pour corriger cette situation, ZFS doit être tenu informé de ne pas rechercher de pool au démarrage.

ZFS conserve un cache interne de pools disponibles et de leurs configurations dans `/etc/zfs/zpool.cache`. L'emplacement et le contenu de ce fichier sont privés et sujets à modification. Si le système devient impossible à initialiser, redémarrez au jalon `none` à l'aide de l'option d'initialisation `-m milestone=none`. Une fois le système rétabli, remontez le système de fichiers root en tant que système accessible en écriture, puis renommez ou placez le fichier `/etc/zfs/zpool.cache` à un autre emplacement. En raison de ces actions, ZFS oublie l'existence de pools dans le système, ce qui l'empêche d'accéder au pool défectueux à l'origine du problème. Vous pouvez ensuite passer à un état normal de système en exécutant la commande `svcadm milestone all`. Vous pouvez utiliser un processus similaire lors de l'initialisation à partir d'un root de remplacement pour effectuer des réparations.

Une fois le système démarré, vous pouvez tenter d'importer le pool à l'aide de la commande `zpool import`. Cependant, dans ce cas, l'erreur qui s'est produite lors de l'initialisation risque de se reproduire car la commande utilise le même mécanisme d'accès aux pools. Si le système contient plusieurs pools, procédez comme suit :

- Renommez ou déplacez le fichier `zpool.cache` vers un autre emplacement comme décrit dans le paragraphe ci-dessus.
- Utilisez la commande `fmdump -eV` pour afficher les pools présentant des erreurs fatales et déterminer ainsi quel pool pose des problèmes.
- Importez les pools un à un en ignorant ceux qui posent problème, comme décrit dans la sortie de la commande `fmdump`.

♦ ♦ ♦ CHAPITRE 11

Pratiques recommandées pour Oracle Solaris ZFS

Ce chapitre décrit les pratiques recommandées pour la création, la surveillance et la gestion de pools de stockage ZFS et de systèmes de fichiers.

Ce chapitre contient les sections suivantes :

Pour plus d'informations sur le réglage ZFS pour une base de données Oracle, reportez-vous auChapitre 3, “ Paramètres réglables ZFS d’Oracle Solaris ” du manuel “ Manuel de référence des paramètres réglables d’Oracle Solaris 11.2 ”

Pratiques recommandées pour les pools de stockage

Les sections suivantes décrivent les pratiques recommandées pour la création et la surveillance de pools de stockage ZFS. Pour plus d'informations sur le dépannage des problèmes de pools de stockage, reportez-vous au Chapitre 10, Dépannage d'Oracle Solaris ZFS et récupération de pool.

Pratiques recommandées générales

- Maintenez votre système à jour grâce aux dernières versions et aux correctifs Solaris.
- Confirmez que votre contrôleur honore les commandes de purge de cache pour être sûr que vos données ont été écrites de manière sécurisée, ce qui permet de modifier les périphériques du pool ou de séparer un pool de stockage mis en miroir. Ce n'est généralement pas un problème sur le matériel Oracle/Sun, mais il est bon de confirmer que les paramètres de purge du cache de votre matériel sont activés.
- Evaluez la mémoire requise en fonction de la charge de travail actuelle du système.
  - Avec une application dont l'encombrement mémoire est connu, telle qu'une application de base de données par exemple, vous pouvez limiter la taille de l'ARC de manière à ce

que l'application n'ait pas besoin de récupérer la mémoire nécessaire à partir du cache ZFS.

- Tenez compte de la mémoire requise pour la suppression des doublons.
- Identifiez l'utilisation de la mémoire par ZFS à l'aide de la commande suivante :

```
# mdb -k
> ::memstat
Page Summary                Pages                MB  %Tot
------------     ----------------  ----------------  ----
Kernel                     388117              1516   19%
ZFS File Data               81321               317    4%
Anon                        29928               116    1%
Exec and libs                1359                 5    0%
Page cache                   4890                19    0%
Free (cachelist)             6030                23    0%
Free (freelist)           1581183              6176   76%

Total                     2092828              8175
Physical                  2092827              8175
> $q
```

- Envisagez l'utilisation de mémoire ECC pour prévenir l'altération de mémoire. L'altération de mémoire silencieuse peut endommager vos données.

- Effectuez des sauvegardes régulières : bien qu'un pool créé avec redondance ZFS permette de réduire le temps d'inactivité dû à des pannes matérielles, il n'est pas à l'abri des pannes matérielles, des pannes d'alimentation ou des déconnexions de câbles. Veillez à sauvegarder régulièrement vos données. Toutes les données importantes doivent être sauvegardées. Il existe différentes méthodes de copie des données :
  - Prise quotidienne ou à intervalles réguliers d'instantanés ZFS.
  - Sauvegardes hebdomadaires des données du pool ZFS. Vous pouvez utiliser la commande `zpool split` pour créer une copie exacte du pool de stockage ZFS mis en miroir.
  - Sauvegardes mensuelles à l'aide d'un produit de sauvegarde mis en oeuvre à l'échelle de l'entreprise.
- RAID matériel
  - Envisagez l'utilisation du mode JBOD pour les baies de stockage plutôt que des baies RAID matérielles, afin que ZFS puisse gérer le stockage et la redondance.
  - Utilisez la redondance matérielle RAID et/ou la redondance ZFS.
  - L'utilisation de la redondance ZFS présente de nombreux avantages : pour les environnements de production, configurez ZFS de manière à lui permettre de réparer les incohérences de données. Utilisez la redondance ZFS, telle que RAID-Z, RAID-Z-2, RAID-Z-3, la mise en miroir, quel que soit le niveau RAID mis en oeuvre sur le périphérique de stockage sous-jacent. Avec une telle redondance, ZFS est en mesure de détecter et de réparer les défaillances du périphérique de stockage sous-jacent ou des connexions à l'hôte de celui-ci.

- Si vous êtes certain de la redondance de votre solution RAID matérielle, envisagez d'utiliser ZFS sans redondance ZFS avec votre baie RAID matérielle. Toutefois, tenez compte des conseils suivants pour vous aider à garantir l'intégrité des données.
  - Affectez des tailles de LUN et de pool de stockage ZFS qui vous semblent raisonnables en considérant que ZFS ne sera pas en mesure de résoudre les incohérences de données si la baie RAID matérielle subit une panne.
  - Créez des LUN RAID5 avec des disques hot spare globaux.
  - Surveillez le pool de stockage ZFS à l'aide de la commande `zpool status` et les LUN sous-jacents à l'aide des outils de surveillance RAID.
  - Remplacez rapidement les périphériques défectueux.
  - Nettoyez régulièrement (par exemple mensuellement) vos pools de stockage ZFS si vous utilisez des services de qualité convenant à un centre de données.
  - Veillez à toujours disposer de sauvegardes de qualité et récentes de vos données importantes.

  Voir aussi “Pratiques recommandées pour la création de pool sur une baie de stockage en réseau ou locale” à la page 326.
- Les vidages sur incident consomment davantage d'espace disque, généralement entre 1/2 et 3/4 de la taille de la mémoire physique.

Pratiques recommandées pour la création de pools ZFS

Les sections suivantes présentent des pratiques recommandées générales et plus spécifiques pour les pools de stockage

Pratiques recommandées générales pour les pools de stockage

- Utilisez des disques entiers pour activer la mise en cache et faciliter la maintenance. La création de pools sur des tranches complique la gestion et la récupération des disques.
- Utilisez la redondance ZFS pour permettre à ZFS de réparer les incohérences de données.
  - Le message suivant s'affiche lorsqu'un pool non redondant est créé :

    ```
    # zpool create tank c4t1d0 c4t3d0
    'tank' successfully created, but with no redundancy; failure
    of one device will cause loss of the pool
    ```
  - Pour des pools mis en miroir, utilisez des paires de disques mis en miroir
  - Pour des pools RAID-Z, regroupez 3 à 9 disques par VDEV

    - Ne mélangez pas les composants RAID-Z et mis en miroir dans un même pool. Ces pools sont plus difficiles à gérer et les performances peuvent en souffrir.
- Utilisez des disques hot spare pour réduire le temps d'inactivité dû aux pannes matérielles.
- Utilisez des disques de taille similaire afin que de répartir les E/S de façon équilibrée entre les périphériques.
    - Des LUN de petite taille peuvent être étendus en LUN de grande taille
    - N'étendez pas des LUN de façon excessive, comme par exemple de 128 Mo à 2 To, de manière à conserver des tailles de metaslabs optimales
- Envisagez la création d'un petit pool root et de pools de données plus volumineux pour assurer une récupération plus rapide du système
- La taille de pool minimale recommandée est de 8 Go. Bien que la taille de pool minimale est de 64 Mo, toute taille inférieure à 8 Go rend l'allocation et la demande d'espace de pool libre plus difficiles.
- La taille de pool maximale recommandée doit être adaptée à votre charge de travail ou à la taille de vos données. N'essayez pas de stocker plus de données que vous ne pouvez en sauvegarder régulièrement. Dans le cas contraire, vos données sont vulnérables en cas d'événement imprévu.

Voir aussi “Pratiques recommandées pour la création de pool sur une baie de stockage en réseau ou locale” à la page 326.

Pratiques recommandées pour la création de pool root

- **SPARC (SMI (VTOC))** : créez des pools root comportant des tranches à l'aide de l'identificateur `s*`. N'utilisez pas l'identificateur `p*`. Le pool root ZFS d'un système est généralement créé au moment de l'installation du système. Si vous êtes en train de créer un second pool root ou de recréer un pool root, utilisez une syntaxe semblable à la suivante :

  ```
  # zpool create rpool c0t1d0s0
  ```

  Sinon, créez un pool root mis en miroir. Par exemple :

  ```
  # zpool create rpool mirror c0t1d0s0 c0t2d0s0
  ```
- **Solaris 11.1 x86 (EFI (GPT))** : créez des pools root sur des disques complets, au moyen de l'identificateur `d*`. N'utilisez pas l'identificateur `p*`. Le pool root ZFS d'un système est généralement créé au moment de l'installation du système. Si vous êtes en train de créer un second pool root ou de recréer un pool root, utilisez une syntaxe semblable à la suivante :

  ```
  # zpool create rpool c0t1d0
  ```

  Sinon, créez un pool root mis en miroir. Par exemple :

  ```
  # zpool create rpool mirror c0t1d0 c0t2d0
  ```
- Le pool root doit être créé sous la forme d'une configuration en miroir ou d'une configuration à disque unique. Les configurations RAID-Z ou entrelacées ne sont pas prises

en charge. Vous ne pouvez pas ajouter d'autres disques mis en miroir pour créer plusieurs périphériques virtuels de niveau supérieur à l'aide de la commande `zpool add`. Toutefois, vous pouvez étendre un périphérique virtuel mis en miroir à l'aide de la commande `zpool attach`.

- Un pool root ne peut pas avoir de périphérique de journal distinct.
- Les propriétés d'un pool peuvent être définies lors d'une installation AI, mais l'algorithme de compression `gzip` n'est pas pris en charge sur les pools root.
- Ne renommez pas le pool root une fois qu'il a été créé par une installation initiale. Si vous renommez le pool root, cela peut empêcher l'initialisation du système.
- Ne créez pas de pool root sur une clé USB pour un système de production, car les disques de pools root sont vitaux pour un fonctionnement continu, en particulier dans un environnement professionnel. Envisagez d'utiliser les disques internes d'un système pour le pool root, ou au moins d'utiliser des disques de la même qualité que celle que vous utiliseriez pour vos données non root. De plus, une clé USB peut s'avérer trop petite pour gérer une taille de fichier de vidage équivalente à la moitié de la taille de la mémoire physique.
- Au lieu d'ajouter un disque hot spare à un pool root, envisagez de créer une à deux ou un miroir tridirectionnel pool root. Ne partagez pas un disque hot spare entre un pool root et un pool de données.
- Thinly, n'utilisez pas ce périphérique VMware pour un pas de privilèges d'accès pour les périphériques de pools root.

Pratiques recommandées pour la création de pools non root

- Créez des pools non root avec des disques entiers à l'aide de l'identificateur `d*`. N'utilisez pas l'identificateur `p*`.
  - ZFS fonctionne mieux sans logiciel de gestion de volumes supplémentaire.
  - Pour de meilleures performances, utilisez des disques individuels ou, tout au moins, des LUN constitués d'un petit nombre de disques. En offrant à ZFS un meilleur aperçu de la configuration des LUN, vous lui permettez de prendre de meilleures décisions de planification d'E/S.
- Créez des configurations de pools redondants dans plusieurs contrôleurs afin de réduire le temps d'inactivité dû à une panne de contrôleur.
  - **Pools de stockage mis en miroir** : consomment davantage d'espace disque mais présentent de meilleures performances pour les petites lectures aléatoires.

    ```
    # zpool create tank mirror c1d0 c2d0 mirror c3d0 c4d0
    ```
  - **Pools de stockage RAID-Z** : ces pools peuvent être créés avec 3 stratégies de parité, d'une parité égale à 1 (`raidz`), 2 (`raidz2`) ou 3 (`raidz3`). Une configuration RAID-Z optimise l'espace disque et généralement effectue bien lorsque les données sont écrites et lues en gros blocs (128 Ko ou plus).

- Prenons l'exemple d'une configuration RAID-Z à parité simple (`raidz`) avec 2 périphériques virtuels à 3 disques (2+1) chacun.

  ```
  # zpool create rzpool raidz1 c1t0d0 c2t0d0 c3t0d0 raidz1 c1t1d0 c2t1d0 c3t1d0
  ```

- Une configuration RAIDZ-2 améliore la disponibilité des données et offre les mêmes performances qu'une configuration RAID-Z. En outre, sa valeur de temps moyen entre pertes de données MTTDL (Mean Time To Data Loss) est nettement meilleure que celle d'une configuration RAID-Z ou de miroirs bidirectionnels. Créez une configuration RAID-Z à double parité RAID-Z (`raidz2`) à 6 disques (4+2).

  ```
  # zpool create rzpool raidz2 c0t1d0 c1t1d0 c4t1d0 c5t1d0 c6t1d0 c7t1d0
  raidz2 c0t2d0 c1t2d0 c4t2d0 c5t2d0 c6t2d0 c7t2d
  ```

- La configuration RAIDZ-3 optimise l'espace disque et offre une excellente disponibilité car elle peut résister à 3 pannes de disque. Créez une configuration RAID-Z à triple parité (`raidz3`) à 9 disques (6+3).

  ```
  # zpool create rzpool raidz3 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0
  c5t0d0 c6t0d0 c7t0d0 c8t0d0
  ```

Pratiques recommandées pour la création de pool sur une baie de stockage en réseau ou locale

Tenez compte des pratiques recommandées pour la création d'un pool root ZFS sur une baie de stockage connectée localement ou à distance.

- Si vous créez un pool sur des périphériques SAN et que la connexion réseau est lente, les périphériques du pool peuvent devenir `UNAVAIL` pendant un certain temps. Vous devez évaluer si la connexion réseau est adéquate pour fournir vos données de manière continue. En outre, considérez, si vous utilisez des périphériques SAN pour votre pool root, ils peuvent ne pas être disponibles dès l'initialisation du système, et les périphériques du pool root peuvent également être `UNAVAIL`.
- Confirmez avec votre vendeur de baies que la baie de stockage ne vide pas son cache après une demande de mise en cache des enregistrements de vidage par ZFS.
- Utilisez des disques entiers, plutôt que des tranches de disque, comme périphériques de pool de stockage afin qu'Oracle Solaris ZFS active les caches des petits disques locaux, qui sont vidés à des moments appropriés.
- Pour de meilleures performances, créez un LUN pour chaque disque physique de la baie. Si vous utilisez un LUN unique et volumineux, ZFS risque de mettre trop peu d'opérations d'E/S de lecture en attente pour effectivement assurer des performances optimales de stockage. A l'inverse, l'utilisation de nombreux petits LUN peut entraîner l'inondation du stockage avec un grand nombre d'opérations d'E/S de lecture en attente.
- Une baie de stockage qui utilise un logiciel d'approvisionnement dynamique (fin) pour implémenter une allocation d'espace virtuel n'est pas recommandée pour Oracle Solaris ZFS. Quand Oracle Solaris ZFS écrit les données modifiées dans l'espace libre, il écrit sur le

LUN entier. Le processus d'écriture d'Oracle Solaris ZFS alloue tout l'espace virtuel depuis le point de vue de la baie de stockage, ce qui annule l'avantage de l'approvisionnement dynamique.

Considérez qu'un logiciel d'approvisionnement dynamique peut s'avérer inutile lors de l'utilisation de ZFS.

- Vous pouvez étendre un LUN dans un pool de stockage ZFS existant et il utilisera le nouvel espace.
- Un comportement similaire fonctionne quand un LUN plus petit est remplacé par un LUN plus gros.
- Si vous évaluez les besoins de stockage pour votre pool et créez le pool avec des LUN plus petits qui correspondent aux besoins de stockage, vous pouvez alors toujours étendre les LUN si vous avez besoin de plus d'espace.

- Si la baie peut présenter des périphériques individuels (mode JBOD), envisagez de créer des pools de stockage ZFS redondants (en miroir ou RAID-Z) sur ce type de baie afin que ZFS puisse signaler et corriger les incohérences de données.

Pratiques recommandées pour la création de pools pour une base de données Oracle

Tenez compte des pratiques recommandées pour la création de pools de stockage suivantes lorsque vous créez une base de données Oracle.

- Utilisez un pool mis en miroir ou un RAID matériel pour plusieurs pools
- Les pools RAID-Z ne sont généralement pas recommandés pour les charges de travail en lecture aléatoire
- Créez un petit pool distinct avec un périphérique de journalisation distinct pour les fichiers de journalisation de la base de données
- Créez un petit pool distinct pour le journal d'archivage

Pour plus d'informations sur le réglage ZFS d'une base de données Oracle, reportez-vous à la section “ Réglage du ZFS pour une base de données Oracle ” du manuel “ Manuel de référence des paramètres réglables d’Oracle Solaris 11.2 ”.

Utilisation de pools de stockage ZFS dans VirtualBox

- Virtual Box est configuré pour ignorer des commandes de vidage de cache à partir du stockage sous-jacent par défaut. Cela signifie que dans le cas d'un blocage système ou d'une défaillance matérielle, les données peuvent être perdues.
- Activez le vidage du cache dans Virtual Box à l'aide de la commande suivante :

```
VBoxManage setextradata vm-name "VBoxInternal/Devices/type/0/LUN#n/Config/IgnoreFlush" 0
```

  - *vm-name* : le nom de la machine virtuelle

- *type* : le type de contrôleur, `piix3ide`, si vous utilisez le contrôleur virtuel IDE habituel, ou `ahci`, si vous utilisez un contrôleur SATA
- *n* – ne numéro de disque

Pratiques recommandées pour l'optimisation des performances des pools de stockage

- N'utilisez pas plus de 90 % de la capacité d'un pool pour des performances optimales.
- Les pools mis en miroir sont à préférer aux pools RAID-Z pour des charges de travail en lecture/écriture aléatoires
- Périphériques de journalisation distincts
  - Recommandés pour améliorer les performances d'écriture synchrone
  - Avec une charge d'écriture synchrone élevée, empêche la fragmentation de l'écriture de nombreux blocs de journal dans le pool principal
- Des périphériques de cache distincts sont recommandés pour améliorer les performances de lecture
- Nettoyage/réargenture : un très grand pool RAID-Z comportant un grand nombre de périphériques nécessite des temps de nettoyage et de réargenture plus long
- Si les performances du pool sont ralenties : utilisez la commande `zpool status` pour éliminer les problèmes matériels à l'origine des problèmes de performances du pool. Si la commande `zpool status` ne fait apparaître aucun problème, utilisez la commande `fmdump` pour afficher les pannes matérielles ou utilisez la commande `fmdump-eV` pour repérer les éventuelles défaillances matérielles qui n'ont pas encore été signalées en tant qu'erreur.

Pratiques recommandées pour la maintenance et la surveillance d'un pool de stockage ZFS

- Assurez-vous que la capacité d'un pool est inférieure à 90 %, pour obtenir de meilleures performances.

  Les performances d'un pool peuvent se dégrader lorsque le pool est très plein et que les systèmes de fichiers sont fréquemment mis à jour, comme c'est le cas par exemple pour un serveur de courrier très actif. Des pools pleins peuvent entraîner une baisse des performances, mais aucun autre problème. Si la charge de travail principale consiste en des fichiers immuables, maintenez un taux d'utilisation du pool de 95 à 96 %. Même avec un contenu essentiellement statique et un taux d'utilisation de 95 à 96 %, les performances d'écriture, de lecture et de réargenture risquent de se dégrader.

  - Surveillez l'espace des pools et des systèmes de fichiers pour vous assurer qu'il n'est pas entièrement utilisé.

- Il vous est conseillé d'envisager l'utilisation quotas et réservations ZFS pour vous assurer que l'espace pour système de fichiers ne dépasse pas 90 % capacité du pool.

- Surveillez la santé du pool
  - Surveillez au moins une fois par semaine un pool redondant avec `zpool status` et `fmdump`.
  - Surveillez au moins deux fois par semaine un pool non redondant avec `zpool status` et `fmdump`
- Exécutez régulièrement `zpool scrub` pour repérer les problèmes d'intégrité des données.
  - Si vous utilisez des unités de qualité grand public, envisagez de planifier un nettoyage hebdomadaire.
  - Si vous utilisez des unités de qualité professionnelle, envisagez de planifier un nettoyage mensuel.
  - Vous devez également exécuter un nettoyage avant de remplacer des périphériques ou de réduire temporairement la redondance d'un pool, afin d'assurer que tous les périphériques sont alors opérationnels.
- Surveillance des défaillances de pools ou de périphériques : utilisez `zpool status` comme décrit ci-dessous. Utilisez également les commandes `fmdump` ou `fmdump -eV` pour vérifier l'absence de défauts et d'erreurs au niveau des périphériques.
  - Surveillez la santé des pools redondants toutes les semaines à l'aide de `zpool status` et `fmdump`
  - Surveillez la santé des pools non redondants deux fois par semaine à l'aide de `zpool status` et `fmdump`
- Le périphérique de pool est `UNAVAIL` ou `OFFLINE` : si un périphérique de pool n'est pas disponible, vérifiez si le périphérique est répertorié dans la sortie de la commande `format`. Si le périphérique n'apparaît pas dans la sortie de `format`, il n'est pas visible sur ZFS.

  L'état de périphérique de pool `UNAVAIL` ou `OFFLINE` signifie généralement que le périphérique est en panne, qu'un câble est déconnecté ou qu'un autre problème matériel, tel qu'un câble ou un contrôleur défectueux, a rendu inaccessible le périphérique.
- Envisagez de configurer le service `smtp-notify` de manière à ce qu'il vous informe lorsqu'un composant matériel est diagnostiqué comme défectueux. Pour plus d'informations, reportez-vous à la rubrique Paramètres de notification des pages de manuel `smf`(5) et `smtp-notify`(1M).

  Par défaut, certaines notifications sont configurées automatiquement pour être envoyées à l'utilisateur root. Si vous ajoutez un alias pour votre compte utilisateur en tant qu'utilisateur root dans le fichier `/etc/aliases`, vous recevrez par courrier électronique des notifications semblables à la suivante :

```
From noaccess@tardis.space.com Fri Jun 29 16:58:59 2012
Date: Fri, 29 Jun 2012 16:58:58 -0600 (MDT)
From: No Access User <noaccess@tardis.space.com>
Message-Id: <201206292258.q5TMwwFL002753@tardis.space.com>
```

```
Subject: Fault Management Event: tardis:ZFS-8000-8A
To: root@tardis.central.com
Content-Length: 771

SUNW-MSG-ID: ZFS-8000-8A, TYPE: Fault, VER: 1, SEVERITY: Critical
EVENT-TIME: Fri Jun 29 16:58:58 MDT 2012
PLATFORM: ORCL,SPARC-T3-4, CSN: 1120BDRCCD, HOSTNAME: tardis
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: 76c2d1d1-4631-4220-dbbc-a3574b1ee807
DESC: A file or directory in pool 'pond' could not be read due to corrupt data.
AUTO-RESPONSE: No automated response will occur.
IMPACT: The file or directory is unavailable.
REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this event.
Run 'zpool status -xv' and examine the list of damaged files to determine what
has been affected. Please refer to the associated reference document at
http://support.oracle.com/msg/ZFS-8000-8A for the latest service procedures
and policies regarding this diagnosis.
```

- Surveillez l'espace du pool de stockage : utilisez les commandes `zpool list` et `zfs list` pour déterminer la quantité d'espace disque utilisée par les données des systèmes de fichiers. Les instantanés ZFS peuvent consommer de l'espace disque et, lorsqu'ils ne sont pas répertoriés par la commande `zfs list`, peuvent consommer de l'espace disque de manière silencieuse. Utilisez la commande d'instantané `zfs list- t` pour identifier l'espace disque consommé par des instantanés.

Pratiques recommandées pour les systèmes de fichiers

Les sections suivantes décrivent les pratiques recommandées pour les systèmes de fichiers.

Pratiques recommandées pour les systèmes de fichiers root

- Envisager de conserver l'de petite taille et de système de fichiers root non root isolé liés à partir d'autres données de manière à ce que la récupération de pool root est plus rapide.
- N'incluez pas les systèmes de fichiers dans rpool / ROOT, qui est un conteneur qui ne nécessite pas d'administration et ne doit contenir aucun des composants supplémentaires.

Pratiques recommandées pour la création de systèmes de fichiers

Les sections suivantes décrivent les pratiques recommandées pour la création de systèmes de fichiers ZFS.

- Créez un système de fichiers par utilisateur pour les répertoires d'accueil
- Envisagez d'utiliser des quotas et des réservations de systèmes de fichiers pour gérer et réserver de l'espace disque pour les systèmes de fichiers importants.
- Envisagez de définir des quotas par utilisateur ou par groupe pour gérer l'espace disque dans un environnement comptant de nombreux utilisateurs.
- Utilisez l'héritage des propriétés ZFS pour appliquer des propriétés à un grand nombre de systèmes de fichiers descendants.

Pratiques recommandées pour la création de systèmes de fichiers pour une base de données Oracle

Tenez compte des pratiques recommandées pour la création de systèmes de fichiers suivantes lorsque vous créez une base de données Oracle.

- Faites correspondre la propriété `recordsize` ZFS à la taille `db_block_size` Oracle.
- Créez des systèmes de fichiers pour la table de base données et l'index dans le pool de base de données principal, en utilisant une taille `recordsize` de 8 Ko et la valeur `primarycache` par défaut.
- Créez des systèmes de fichiers pour les données temporaires et l'espace de la table d'annulation dans le pool de base de données principal, en utilisant les valeurs `recordsize` et `primarycache` par défaut.
- Créez un système de fichiers pour le journal d'archivage dans le pool d'archivage, en activant la compression et la valeur `recordsize` par défaut et en définissant `primarycache` sur `metadata`.

Pour plus d'informations, consultez le livre blanc suivant :

http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle

Pratiques recommandées pour la surveillance de systèmes de fichiers ZFS

Il est recommandé de surveiller les systèmes de fichiers ZFS pour s'assurer qu'ils sont disponibles et pour identifier les problèmes de consommation d'espace.

- Une fois par semaine, contrôlez l'espace disponible des systèmes de fichiers à l'aide des commandes `zpool list` et `zfs list` au lieu des commandes `du` et `df`, car les anciennes commandes ne tiennent pas compte de l'espace consommé par les systèmes de fichiers descendants ou les instantanés.

  Pour plus d'informations, reportez-vous à la section “Résolution des problèmes d'espace ZFS” à la page 308.
- Affichez la consommation d'espace des systèmes de fichiers à l'aide de la commande `zfs list -o space`.
- L'espace des systèmes de fichiers peut être consommé par des instantanés à votre insu. Vous pouvez afficher toutes les informations sur les systèmes de fichiers en respectant la syntaxe suivante :

  ```
  # zfs list -t all
  ```
- Un système de fichiers `/var` distinct est automatiquement créé lorsqu'un système est installé, mais il est recommandé de définir un quota et une réservation sur ce système de fichiers pour s'assurer qu'il ne consomme pas d'espace de pool root à votre insu.
- En outre, vous pouvez utiliser la commande `fsstat` pour afficher les activités de traitement de fichiers des systèmes de fichiers ZFS. Les activités peuvent être consignées par point de montage ou par type de système de fichiers. L'exemple suivant illustre les activités générales de système de fichiers ZFS :

  ```
  # fsstat /
  new  name   name  attr  attr lookup rddir  read read  write write
  file remov  chng   get   set    ops   ops   ops bytes   ops bytes
  832   589   286  837K 3.23K  2.62M 20.8K 1.15M 1.75G 62.5K  348M /
  ```
- Sauvegardes
  - Conservez des instantanés des systèmes de fichiers
  - Envisagez d'effectuer des sauvegardes hebdomadaires et mensuelles à l'aide de logiciels mis en oeuvre à l'échelle de l'entreprise
  - Stockez des instantanés des pools root sur un système distant pour la récupération à chaud

♦ ♦ ♦ ANNEXE A

Description 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

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 les versions des pools et des systèmes de fichiers.

Pour plus d'informations sur l'utilisation des commandes `zpool upgrade` et `zfs upgrade`, reportez-vous aux sections "Mise à niveau des systèmes de fichiers ZFS" à la page 202 et "Mise à niveau de pools de stockage ZFS" à la page 101.

Versions de pool ZFS

Le tableau suivant présente une liste des versions de pool ZFS disponibles dans la version d'Oracle Solaris.

version	Oracle Solaris 11	Description
1	snv_36	Version ZFS initiale
2	snv_38	Blocs ditto (métadonnées répliquées)

version	Oracle Solaris 11	Description
3	snv_42	Disques hot spare et RAID-Z à double parité
4	snv_62	`zpool history`
5	snv_62	Algorithme de compression `gzip`
6	snv_62	Propriété de pool `bootfs`
7	snv_68	Périphérique de journalisation de tentatives distincts
8	snv_69	Administration déléguée
9	snv_77	Propriétés `refquota` et `refreservation`
10	snv_78	Périphériques de cache
11	snv_94	Performances de nettoyage améliorées
12	snv_96	Propriétés d'instantané
13	snv_98	Propriété `snapused`
14	snv_103	Propriété `Aclinherit` `passthrough-x`
15	snv_114	Comptabilisation des espaces de groupe et d'utilisateur
16	snv_116	Propriété `stmf`
17	snv_120	RAID-Z triple parité
18	snv_121	Instantanés conservés par l'utilisateur
19	snv_125	Suppression d'un périphérique de journalisation
20	snv_128	Algorithme de compression `zle` (encodage de chaîne vide)
21	snv_128	Suppression des doublons
22	snv_128	Propriétés reçues
23	snv_135	Slim ZIL
24	snv_137	Attributs système
25	snv_140	Statistiques de nettoyage améliorées
26	snv_141	Performances de suppression des instantanés améliorées
27	snv_145	Performances de création d'instantanés améliorées
28	snv_147	Remplacements de vdev multiples
29	snv_148	Programme d'allocation hybride de miroir/RAID-Z
30	snv_149	Chiffrement
31	snv_150	Performances "zfs list" améliorées
32	snv_151	Taille de bloc d'1 Mo
33	snv_163	Prise en charge améliorée du partage
34	S11.1	Partage avec héritage

Versions du système de fichiers ZFS

Le tableau suivant répertorie les versions du système de fichiers ZFS disponibles dans la version d'Oracle Solaris. N'oubliez pas qu'une fonction disponible dans certaines versions du système de fichiers nécessite une version de pool spécifique.

version	Oracle Solaris 11	Description
1	snv_36	Version initiale du système de fichiers ZFS
2	snv_69	Entrées de répertoire améliorées
3	snv_77	Insensibilité à la casse et identificateur unique de système de fichiers (FUID)
4	snv_114	Propriétés `userquota` et `groupquota`
5	snv_137	Attributs système
6	S11.1	Prise en charge de système de fichiers multiniveau

Index

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

X

Z