

Guia de administração do ZFS Oracle Solaris

Copyright © 2006, 2010, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados e de titularidade da Oracle Corporation. Proibida a reprodução total ou parcial.

Este programa de computador e sua documentação são fornecidos sob um contrato de licença que contém restrições sobre seu uso e divulgação, sendo também protegidos pela legislação de propriedade intelectual. Exceto em situações expressamente permitidas no contrato de licença ou por lei, não é permitido usar, reproduzir, traduzir, divulgar, modificar, licenciar, transmitir, distribuir, expor, executar, publicar ou exibir qualquer parte deste programa de computador e de sua documentação, de qualquer forma ou através de qualquer meio. Não é permitida a engenharia reversa, a desmontagem ou a descompilação deste programa de computador, exceto se exigido por lei para obter interoperabilidade.

As informações contidas neste documento estão sujeitas a alteração sem aviso prévio. A Oracle Corporation não garante que tais informações estejam isentas de erros. Se você encontrar algum erro, por favor, nos envie uma descrição de tal problema por escrito.

Se este programa de computador, ou sua documentação, for entregue / distribuído(a) ao Governo dos Estados Unidos ou a qualquer outra parte que licencie os Programas em nome daquele Governo, a seguinte nota será aplicável:

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

Este programa de computador foi desenvolvido para uso em diversas aplicações de gerenciamento de informações. Ele não foi desenvolvido nem projetado para uso em aplicações inerentemente perigosas, incluindo aquelas que possam criar risco de lesões físicas. Se utilizar este programa em aplicações perigosas, você será responsável por tomar todas e quaisquer medidas apropriadas em termos de segurança, backup e redundância para garantir o uso seguro de tais programas de computador. A Oracle Corporation e suas afiliadas se isentam de qualquer responsabilidade por quaisquer danos causados pela utilização deste programa de computador em aplicações perigosas.

Oracle e Java são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

AMD, Opteron, o logotipo da AMD e o logotipo do AMD Opteron são marcas comerciais ou marcas comerciais registradas da Advanced Micro Devices. Intel e Intel Xeon são marcas comerciais ou marcas comerciais registradas da Intel Corporation. Todas as marcas comerciais SPARC são usadas sob licença e são marcas comerciais ou marcas comerciais registradas da SPARC International, Inc. UNIX é uma marca comercial registrada licenciada por meio do consórcio X/Open Company Ltd.

Este programa e sua documentação podem oferecer acesso ou informações relativas a conteúdos, produtos e serviços de terceiros. A Oracle Corporation e suas empresas afiliadas não fornecem quaisquer garantias relacionadas a conteúdos, produtos e serviços de terceiros e estão isentas de quaisquer responsabilidades associadas a eles. A Oracle Corporation e suas empresas afiliadas não são responsáveis por quaisquer tipos de perdas, despesas ou danos incorridos em consequência do acesso ou da utilização de conteúdos, produtos ou serviços de terceiros.

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

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

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

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

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

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

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

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

Conteúdo

Prefácio	11
1 Sistema de arquivos Oracle Solaris ZFS (introdução)	17
O que há de novo no ZFS?	17
Dividindo um conjunto de armazenamento do ZFS espelhado (<code>zpool split</code>)	18
Novo processo do sistema ZFS	19
Alterações no comando <code>zpool list</code>	19
Recuperação do conjunto de armazenamento do ZFS	19
Aprimoramentos no dispositivo de log do ZFS	20
RAIDZ de paridade tripla (<code>raidz3</code>)	20
Mantendo instantâneos do ZFS	20
Aprimoramentos de substituição de dispositivo do ZFS	21
Suporte de instalação do ZFS e Flash	22
Cotas de usuário e grupo do ZFS	23
Herança ACL de passar para permissões de executar ZFS	24
Aprimoramentos de propriedades do ZFS	24
Recuperação do dispositivo de registro do ZFS	27
Utilizando dispositivos de cache no conjunto de armazenamento do ZFS	28
Migração de região em um ambiente do ZFS	29
Suporte à inicialização e instalação do ZFS	29
Revertendo um conjunto de dados sem desmontar	29
Aprimoramentos no comando <code>zfs send</code>	29
Cotas e reservas do ZFS somente para dados do sistema de arquivos	30
Propriedades do pool de armazenamento do ZFS	31
Aprimoramentos no histórico do comando do ZFS (<code>zpool history</code>)	32
Atualizando os sistemas de arquivos do ZFS (<code>zfs upgrade</code>)	33
Administração delegada do ZFS	33
Configurando dispositivos de log do ZFS separados	33

Criando conjuntos de dados do ZFS intermediários	35
Aprimoramentos de conexão automática do ZFS	35
Renomear recursivamente instantâneos do ZFS (zfs rename -r)	36
gzip A compactação está disponível para ZFS	37
Armazenamento de várias cópias de dados de usuário do ZFS	37
Saída zpool status aprimorada	38
Aprimoramentos na iSCSI do Solaris e ZFS	38
Histórico do comando ZFS (zpool history)	39
Melhorias das propriedades do ZFS	40
Exibindo todas as informações do sistema de arquivos do ZFS	40
Nova opção zfs receive -F	41
Instantâneos recursivos do ZFS	41
RAID-Z de paridade dupla (raidz2)	41
Sobressalentes para dispositivos do pool de armazenamento do ZFS	42
Substituindo um sistema de arquivos do ZFS por um clone do ZFS (zfs promote)	42
Atualizando conjuntos de armazenamento do ZFS (zpool upgrade)	42
Comandos de restauração e backup do ZFS renomeados	43
Recuperando conjuntos de armazenamento destruídos	43
O ZFS é integrado com o gerenciador de falhas	43
O comando zpool clear	44
Formato compacto da ACL NFSv4	44
Ferramenta de monitoramento do sistema de arquivos (fsstat)	44
Gerenciamento do ZFS baseado na Web	44
O que é ZFS?	45
Armazenamento de ZFS em pool	45
Semânticas transacionais	46
Somos de verificação e autocorreção de dados	47
Escalabilidade não paralela	47
Instantâneos do ZFS	47
Administração simplificada	47
Terminologia do ZFS	48
Requisitos para nomeação de componentes do ZFS	50
2 Introdução ao ZFS do Oracle Solaris	53
Recomendações e requisitos de software e hardware do ZFS	53

Criando um sistema de arquivos ZFS básico	54
Criando um pool de armazenamento do ZFS	55
▼ Como identificar os requisitos de armazenamento para o pool de armazenamento do ZFS	55
▼ Como criar um pool de armazenamento do ZFS	55
Criando uma hierarquia de sistemas de arquivos ZFS	56
▼ Como determinar a hierarquia do sistema de arquivos ZFS	57
▼ Como criar sistemas de arquivos ZFS	57
3 Diferenças entre o sistema de arquivos tradicional e o ZFS do Oracle Solaris	61
Granularidade do sistema de arquivos ZFS	61
Contabilidade de espaço em disco do ZFS	62
Comportamento por espaço excedido	62
Montando sistemas de arquivos ZFS	63
Gerenciamento de volume tradicional	63
Novo modelo de ACL do Solaris	63
4 Gerenciando conjuntos de armazenamento ZFS do Oracle Solaris	65
Componentes de um pool de armazenamento do ZFS	65
Usando discos em um pool de armazenamento do ZFS	65
Usando faixas em um pool de armazenamento do ZFS	67
Usando arquivos em um pool de armazenamento do ZFS	68
Recursos de replicação de um pool de armazenamento do ZFS	69
Configuração de pool de armazenamento espelhado	69
Configuração de pool de armazenamento RAID-Z	69
Pool armazenamento híbrido do ZFS	71
Dados de autocorreção em uma configuração redundante	71
Distribuição dinâmica em um pool de armazenamento	71
Criando e destruindo pools de armazenamento do ZFS	72
Criando um pool de armazenamento do ZFS	72
Exibindo informações do dispositivo virtual do pool de armazenamento	77
Manipulando erros de criação do pool de armazenamento do ZFS	78
Destruindo pools de armazenamento do ZFS	81
Gerenciando dispositivos em pools de armazenamento do ZFS	82
Adicionando dispositivos a um pool de armazenamento	82

Anexando e desanexando dispositivos em um pool de armazenamento	87
Criando um novo conjunto ao dividir um conjunto de armazenamento ZFS espelhado ...	89
Colocando dispositivos off-line e on-line em um pool de armazenamento	92
Limpando erros de dispositivo de conjunto de armazenamento	95
Substituindo dispositivos em um pool de armazenamento	95
Designando sobressalentes no pool de armazenamento	97
Gerenciando propriedades do pool de armazenamento do ZFS	103
Consultando status de pool de armazenamento do ZFS	106
Exibindo informações sobre os conjuntos de armazenamento ZFS	106
Visualizando estatísticas E/S para conjuntos de armazenamento ZFS	110
Determinando o status de integridade de pools de armazenamento do ZFS	112
Migrando pools de armazenamento do ZFS	115
Preparando para migração do pool de armazenamento do ZFS	115
Exportando um pool de armazenamento do ZFS	116
Determinando pools de armazenamento disponíveis para importação	116
Importando conjuntos de armazenamento ZFS de diretórios alternativos	118
Importando pools de armazenamento do ZFS	119
Recuperando pools de armazenamento destruídos do ZFS	120
Atualizando pools de armazenamento do ZFS	121
5 Instalando e inicializando um sistema de arquivos raiz ZFS do Oracle Solaris	125
Instalando e inicializando um sistema de arquivos ZFS do Oracle Solaris (visão geral)	126
Recursos de instalação do ZFS	126
Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS	127
Instalando um sistema de arquivos raiz ZFS (Instalação inicial)	130
▼ Como criar um conjunto raiz espelhado (pós-instalação)	136
Instalando um sistema de arquivos raiz ZFS (instalação de arquivo do Oracle Solaris Flash) .	137
Instalando um sistema de arquivos raiz ZFS (instalação Oracle Solaris JumpStart)	139
Palavras-chave do JumpStart para ZFS	140
Exemplos de perfil do JumpStart para ZFS	142
Problemas do JumpStart ZFS	142
Migrando um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS (Oracle Solaris Live Upgrade)	143
Problemas de migração ZFS com o Oracle Solaris Live Upgrade	144
Utilizando o Oracle Solaris Live Upgrade para migrar para um sistema de arquivos raiz ZFS	

(sem regiões)	145
Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (Solaris 10 10/08)	149
Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (no mínimo Solaris 10 5/09)	155
Suporte do ZFS para dispositivos de intercâmbio e dump	165
Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS	166
Solução de problemas para dispositivo de despejo ZFS	168
Inicializando a partir de um sistema de arquivos raiz ZFS	169
Inicializando a partir de um disco alternativo em um pool raiz ZFS espelhado	169
SPARC: inicializando a partir de um sistema de arquivos raiz ZFS	170
x86: inicializando a partir de um sistema de arquivos raiz ZFS	172
Resolvendo problemas do ponto de montagem ZFS que impedem a inicialização com êxito (Solaris 10 10/08)	173
Inicializando com o propósito de recuperação em um ambiente raiz ZFS	175
Recuperando o pool raiz ZFS ou instantâneos do pool raiz	176
▼ Como substituir um disco no pool raiz ZFS	176
▼ Como criar instantâneos do pool raiz	179
▼ Como recriar um pool raiz ZFS e recuperar instantâneos do pool raiz	180
▼ Como reverter os instantâneos do pool raiz a partir da inicialização à prova de falhas	182
6 Gerenciando sistemas de arquivos ZFS do Oracle Solaris	183
Gerenciando de sistemas de arquivos ZFS (visão geral)	183
Criando, destruindo e renomeando sistemas de arquivos ZFS	184
Criando um sistema de arquivos ZFS	184
Destruindo um sistema de arquivos ZFS	185
Renomeando um sistema de arquivos ZFS	186
Introduzindo as propriedades do ZFS	187
Propriedades nativas somente leitura do ZFS	195
Propriedades nativas definíveis do ZFS	197
Propriedades de usuário do ZFS	200
Consultando informações sobre o sistema de arquivos ZFS	201
Listando informações básicas do ZFS	201
Criando consultas complexas de ZFS	202
Gerenciando propriedades do ZFS	203
Definindo propriedades do ZFS	203

Herdando propriedades do ZFS	204
Consultando propriedades do ZFS	205
Montando e compartilhando sistemas de arquivos ZFS	208
Gerenciando pontos de montagem do ZFS	208
Montando sistemas de arquivos ZFS	210
Usando propriedades de montagem temporárias	212
Desmontando sistemas de arquivos	212
Compartilhando e descompartilhando sistemas de arquivos ZFS	213
Definindo cotas e reservas do ZFS	215
Definindo cotas em sistemas de arquivos ZFS	216
Definindo reservas nos sistemas de arquivos ZFS	219
7 Trabalhando com instantâneos e clones do ZFS do Oracle Solaris	221
Visão geral dos instantâneos do ZFS	221
Criando e destruindo instantâneos do ZFS	222
Exibindo e acessando instantâneos do ZFS	225
Retornando um instantâneo ZFS	227
Visão geral dos clones do ZFS	228
Criando um clone do ZFS	228
Destruindo um clone do ZFS	229
Substituindo um sistema de arquivos ZFS por um clone do ZFS	229
Enviando e recebendo dados do ZFS	230
Salvando dados do ZFS com outros produtos de backup	231
Enviando um instantâneo do ZFS	231
Recebendo um instantâneo do ZFS	232
Enviando e recebendo fluxos complexos de instantâneos do ZFS	233
8 Utilizando ACLs para proteger arquivos ZFS do Oracle Solaris	237
Novo modelo de ACL do Solaris	237
Descrições de sintaxe para definição de ACLs	239
Herança da ACL	242
Propriedades da ACL	242
Definindo ACLs em arquivos ZFS	243
Definindo e exibindo ACLs em arquivos ZFS no formato verboso	246
Definindo a herança da ACL em arquivos ZFS no formato verboso	251

Definindo e exibindo ACLs em arquivos ZFS no formato compacto	258
9 Administração delegada de ZFS	265
Visão geral da administração delegada de ZFS	265
Desativando permissões delegadas do ZFS	266
Delegando permissões do ZFS	266
Delegação de permissões do ZFS (zfs allow)	268
Removendo as permissões delegadas do ZFS (zfs unallow)	269
Usando a administração delegada de ZFS	270
Delegando permissões do ZFS (Exemplos)	270
Exibindo permissões delegadas do ZFS (Exemplos)	274
Removendo permissões do ZFS (Exemplos)	275
10 Tópicos avançados do ZFS do Oracle Solaris	277
Volumes ZFS	277
Usando um volume ZFS como um dispositivo de intercâmbio ou de dump	278
Usando um volume ZFS como um destino iSCSI Solaris	279
Usando ZFS em um sistema Solaris com regiões instaladas	280
Adicionando sistemas de arquivos ZFS a uma região não global	281
Delegando conjuntos de dados a uma região não global	282
Adicionando volumes ZFS a uma região não global	282
Usando pools de armazenamento do ZFS dentro de uma região	283
Gerenciando propriedades do ZFS dentro de uma região	283
Entendendo a propriedade zoned	284
Usando pools de raiz alternativa do ZFS	286
Criando pools de raiz alternativa do ZFS	286
Importando pools de raiz alternativa	286
Perfis de direitos do ZFS	287
11 Solução de problemas e conjunto de recuperação do Oracle Solaris ZFS	289
Identificando falhas do ZFS	289
Ausência de dispositivos em um pool de armazenamento do ZFS	290
Dispositivos danificados em um pool de armazenamento do ZFS	290
Dados corrompidos do ZFS	290

Verificando a integridade do sistema de arquivos ZFS	291
Reparo de sistema de arquivos	291
Validação do sistema de arquivos	291
Controlando o scrubbing de dados do ZFS	292
Resolvendo problemas com o ZFS	293
Determinando se há problemas em um conjunto de armazenamento do ZFS	294
Revisando a saída de <code>zpool status</code>	295
Relatório de mensagens de erros do ZFS do sistema	298
Reparando uma configuração do ZFS danificada	298
Reparando um dispositivo faltando	299
Reanexando fisicamente um dispositivo	300
Notificando o ZFS da disponibilidade de um dispositivo	300
Substituindo ou reparando um dispositivo modificado	301
Determinando o tipo de falha do dispositivo	301
Apagando erros transitórios	303
Substituindo um dispositivo em um pool de armazenamento do ZFS	303
Reparando dados danificados	310
Identificando o tipo de corrupção de dados	311
Reparando arquivos ou diretórios corrompidos	312
Reparando o dano de todo o pool de armazenamento do ZFS	313
Reparando um sistema não inicializável	314
A Descrição da versão do ZFS do Oracle Solaris	317
Visão geral das versões do ZFS	317
Versões do conjunto do ZFS	317
Versões do sistema de arquivos do ZFS	318
Índice	321

Prefácio

O *Guia de administração do ZFS do Oracle Solaris* fornece informações sobre a configuração e o gerenciamento de sistemas de arquivos ZFS do Oracle Solaris.

Este guia contém informações para sistemas baseados em SPARC e em x86.

Observação – Esta versão do Oracle Solaris oferece suporte a sistemas que utilizam as famílias SPARC e x86 de arquiteturas de processadores: UltraSPARC, SPARC64, AMD64, Pentium e Xeon EM64T. Os sistemas aceitos aparecem na *Lista de compatibilidade de hardware do Solaris 10* em <http://www.sun.com/bigadmin/hcl>. Este documento cita quaisquer diferenças de implementação entre os tipos de plataformas.

Neste documento, os termos x86 apresentam o seguinte significado:

- "x86" refere-se à família maior de produtos compatíveis x86 de 64 bits e de 32 bits.
- "x64" aponta as informações específicas sobre os sistemas AMD64 ou EM64T.
- "32-bit x86" aponta informações específicas sobre os sistemas de 32 bits com base no x86.

Para obter os sistemas suportados, consulte a *Lista de Compatibilidade de Hardware Solaris*.

Quem deve usar este livro

Este guia destina-se àqueles que se interessam pela configuração e gerenciamento dos sistemas de arquivos ZFS Oracle Solaris. Recomenda-se ter experiência com o sistema operacional (OS) Oracle Solaris ou outra versão do UNIX.

Como este livro é organizado

O quadro abaixo descreve os capítulos deste livro.

Capítulo	Descrição
Capítulo 1, “Sistema de arquivos Oracle Solaris ZFS (introdução)”	Oferece uma visão geral do ZFS e de seus recursos e benefícios. Abrange também algumas terminologias e conceitos básicos.
Capítulo 2, “Introdução ao ZFS do Oracle Solaris”	Oferece instruções passo a passo sobre as configurações básicas do ZFS com conjuntos e sistemas de arquivos simples. Este capítulo oferece também o hardware e o software necessários para criar sistemas de arquivos ZFS.
Capítulo 3, “Diferenças entre o sistema de arquivos tradicional e o ZFS do Oracle Solaris”	Identifica os importantes recursos que tornam o ZFS significativamente diferente dos sistemas de arquivos tradicionais. A compreensão dessas diferenças chave o ajudará a reduzir a confusão ao utilizar ferramentas tradicionais para interagir com o ZFS.
Capítulo 4, “Gerenciando conjuntos de armazenamento ZFS do Oracle Solaris”	Oferece uma descrição detalhada de como criar e administrar conjuntos de armazenamento do ZFS.
Capítulo 5, “Instalando e inicializando um sistema de arquivos raiz ZFS do Oracle Solaris”	Descreve como instalar e inicializar um sistema de arquivos ZFS Também trata da migração de um sistema de arquivos raiz do UFS para um sistema de arquivos raiz do ZFS utilizando o Oracle Solaris Live Upgrade.
Capítulo 6, “Gerenciando sistemas de arquivos ZFS do Oracle Solaris”	Oferece informações detalhadas sobre o gerenciamento de sistemas de arquivos ZFS. Estão incluídos conceitos como layout do sistema de arquivos hierárquico, herança de propriedade, gerenciamento do ponto de montagem automático e interações compartilhadas.
Capítulo 7, “Trabalhando com instantâneos e clones do ZFS do Oracle Solaris”	Descreve como criar e administrar clones e instantâneos do ZFS.
Capítulo 8, “Utilizando ACLs para proteger arquivos ZFS do Oracle Solaris”	Descreve como utilizar as listas de controle de acesso (ACLs) para proteger os arquivos do ZFS, proporcionando permissões mais granulares que as permissões padrão do UNIX.
Capítulo 9, “Administração delegada de ZFS”	Descreve como utilizar a administração delegada do ZFS para permitir que usuários não-privilegiados efetuem tarefas de administração do ZFS.
Capítulo 10, “Tópicos avançados do ZFS do Oracle Solaris”	Fornecer informações sobre o uso dos volumes do ZFS, sobre o uso do ZFA em um sistema Oracle Solaris com regiões instaladas e conjuntos raiz alternativas.
Capítulo 11, “Solução de problemas e conjunto de recuperação do Oracle Solaris ZFS”	Descreve como identificar falhas do ZFS e como realizar recuperações a partir delas. Abrange também medidas para evitar falhas.
Apêndice A, “Descrição da versão do ZFS do Oracle Solaris”	Descreve versões do ZFS disponíveis, recursos de cada versão, e o Solaris OS que fornece a versão do ZFS e o recurso.

Livros relacionados

Informações relacionadas aos tópicos gerais de administração do sistema Oracle Solaris podem ser encontradas nos seguintes livros:

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

Documentação, suporte e treinamento

Consulte os sites da Web a seguir para obter recursos adicionais:

- Documentação (<http://docs.sun.com>)
- Suporte (<http://www.oracle.com/us/support/systems/index.html>)
- Treinamento (<http://education.oracle.com>): clique no link da Sun na barra de navegação esquerda.

A Oracle agradece seus comentários

A Oracle agradece seus comentários e sugestões sobre a qualidade e utilidade da documentação. Se encontrar erros ou tiver sugestões para melhorias, vá para <http://docs.sun.com> e clique em Comentários. Indique o título e o número da parte da documentação junto com o capítulo, seção e número da página, se disponíveis. Informe-nos se desejar uma resposta.

Rede de tecnologia Oracle (<http://www.oracle.com/technetwork/index.html>) oferece uma gama de recursos relacionados ao software da Oracle:

- Discuta problemas e soluções técnicas nos Fóruns de discussão (<http://forums.oracle.com>).
- Participe de tutoriais passo-a-passo com o Oracle por exemplo (<http://www.oracle.com/technology/obe/start/index.html>).
- Faça o download do Código de amostra (http://www.oracle.com/technology/sample_code/index.html).

Convenções tipográficas

A tabela a seguir descreve as convenções tipográficas usadas neste livro.

TABELA P-1 Convenções tipográficas

Fonte	Significado	Exemplo
AaBbCc123	Os nomes de comandos, arquivos e diretórios e saída para computador na tela	Edite o arquivo <code>.login</code> . Use <code>ls -a</code> para listar todos os arquivos. <code>nome_da_máquina% you have mail.</code>
AaBbCc123	O que você digitou, contrastado com a saída para computador na tela	<code>nome_da_máquina% su</code> Senha:
<i>aabbcc123</i>	Espaço reservado: substituir por um nome ou valor real	O comando para remover um arquivo é <code>rm nome do arquivo</code> .
AaBbCc123	Títulos de livro, novos termos e termos a serem enfatizados	Leia o Capítulo 6 do <i>Guia do Usuário</i> . Um <i>cache</i> é uma cópia que é armazenada localmente. <i>Não</i> salve o arquivo. Nota: Alguns itens enfatizados aparecem on-line em negrito.

Prompts do shell em exemplos de comando

A tabela a seguir mostra o prompt do sistema UNIX padrão e o prompt do superusuário para shells incluídas no Oracle Solaris OS. Observe que o prompt do sistema padrão que é exibido em exemplos de comando varia, dependendo na versão do Oracle Solaris.

TABELA P-2 Prompts do shell

Shell	Prompt
Bash shell, Korn shell, e Bourne shell	\$
Bash shell, Korn shell, e Bourne shell para superusuário	#
Shell C	nome_da_máquina%
Shell C para superusuário	nome_da_máquina#

Sistema de arquivos Oracle Solaris ZFS (introdução)

Este capítulo oferece uma visão geral do sistema de arquivos Oracle Solaris ZFS e de seus recursos e benefícios. Este capítulo abrange também algumas terminologias básicas usadas neste livro.

Este capítulo traz as seguintes seções:

- “O que há de novo no ZFS?” na página 17
- “O que é ZFS?” na página 45
- “Terminologia do ZFS” na página 48
- “Requisitos para nomeação de componentes do ZFS” na página 50

O que há de novo no ZFS?

Esta seção resume os novos recursos do sistema de arquivos ZFS.

- “Dividindo um conjunto de armazenamento do ZFS espelhado (`zpool split`)” na página 18
- “Novo processo do sistema ZFS” na página 19
- “Alterações no comando `zpool list`” na página 19
- “Recuperação do conjunto de armazenamento do ZFS” na página 19
- “Aprimoramentos no dispositivo de log do ZFS” na página 20
- “RAIDZ de paridade tripla (`raidz3`)” na página 20
- “Mantendo instantâneos do ZFS” na página 20
- “Aprimoramentos de substituição de dispositivo do ZFS” na página 21
- “Suporte de instalação do ZFS e Flash” na página 22
- “Cotas de usuário e grupo do ZFS” na página 23
- “Herança ACL de passar para permissões de executar ZFS” na página 24
- “Aprimoramentos de propriedades do ZFS” na página 24
- “Recuperação do dispositivo de registro do ZFS” na página 27
- “Utilizando dispositivos de cache no conjunto de armazenamento do ZFS” na página 28
- “Migração de região em um ambiente do ZFS” na página 29

- “Suporte à inicialização e instalação do ZFS” na página 29
- “Revertendo um conjunto de dados sem desmontar” na página 29
- “Aprimoramentos no comando `zfs send`” na página 29
- “Cotas e reservas do ZFS somente para dados do sistema de arquivos” na página 30
- “Propriedades do pool de armazenamento do ZFS” na página 31
- “Aprimoramentos no histórico do comando do ZFS (`zpool history`)” na página 32
- “Atualizando os sistemas de arquivos do ZFS (`zfs upgrade`)” na página 33
- “Administração delegada do ZFS” na página 33
- “Configurando dispositivos de log do ZFS separados” na página 33
- “Criando conjuntos de dados do ZFS intermediários” na página 35
- “Aprimoramentos de conexão automática do ZFS” na página 35
- “Renomear recursivamente instantâneos do ZFS (`zfs rename -r`)” na página 36
- “`gzip` A compactação está disponível para ZFS” na página 37
- “Armazenamento de várias cópias de dados de usuário do ZFS” na página 37
- “Saída `zpool status` aprimorada” na página 38
- “Aprimoramentos na iSCSI do Solaris e ZFS” na página 38
- “Histórico do comando ZFS (`zpool history`)” na página 39
- “Melhorias das propriedades do ZFS” na página 40
- “Exibindo todas as informações do sistema de arquivos do ZFS” na página 40
- “Nova opção `zfs receive -F`” na página 41
- “Instantâneos recursivos do ZFS” na página 41
- “RAID-Z de paridade dupla (`raidz2`)” na página 41
- “Sobressalentes para dispositivos do pool de armazenamento do ZFS” na página 42
- “Substituindo um sistema de arquivos do ZFS por um clone do ZFS (`zfs promote`)” na página 42
- “Atualizando conjuntos de armazenamento do ZFS (`zpool upgrade`)” na página 42
- “Comandos de restauração e backup do ZFS renomeados” na página 43
- “Recuperando conjuntos de armazenamento destruídos” na página 43
- “O ZFS é integrado com o gerenciador de falhas” na página 43
- “O comando `zpool clear`” na página 44
- “Formato compacto da ACL NFSv4” na página 44
- “Ferramenta de monitoramento do sistema de arquivos (`fsstat`)” na página 44
- “Gerenciamento do ZFS baseado na Web” na página 44

Dividindo um conjunto de armazenamento do ZFS espelhado (`zpool split`)

Versão 10 9/10 do Oracle Solaris: nessa versão do Solaris, você pode utilizar o comando `zpool split` para dividir um conjunto de armazenamento espelhado, o que desanexa um disco ou discos do conjunto espelhado original para criar outro conjunto idêntico.

Para maiores informações, consulte “Criando um novo conjunto ao dividir um conjunto de armazenamento ZFS espelhado” na página 89.

Novo processo do sistema ZFS

Versão 10 9/10 do Oracle Solaris: nessa versão do Solaris, cada conjunto de armazenamento do ZFS possui um processo associado, `zpool - poolname`. Os segmentos nesse processo são os segmentos de processamento de E/S do conjunto que manipulam tarefas de E/S tais como compactação e soma de verificação, que estão associadas ao conjunto. A finalidade desse processo é oferecer visibilidade em cada utilização de CPU do conjunto de armazenamento. Informações sobre esse processo podem ser revisadas utilizando os comandos `ps` e `prstat`. Esses processos estão disponíveis apenas na região global. Para obter mais informações, consulte [SDC\(7\)](#).

Alterações no comando `zpool list`

Versão 10 9/10 do Oracle Solaris: nessa versão do Solaris, a saída `zpool list` mudou para oferecer melhores informações de alocação de espaço. Por exemplo:

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

Os campos `USED` e `AVAIL` anteriores foram substituídos por `ALLOC` e `FREE`.

O campo `ALLOC` identifica a quantidade de espaço físico alocado para todos os conjuntos de dados e metadados internos. O campo `FREE` identifica a quantidade de espaço não alocada no conjunto.

Para maiores informações, consulte, [“Exibindo informações sobre os conjuntos de armazenamento ZFS”](#) na página 106.

Recuperação do conjunto de armazenamento do ZFS

Versão 10 9/10 do Oracle Solaris: um conjunto de armazenamento pode ficar danificado se dispositivos subjacentes ficarem indisponíveis, se ocorrer uma falha de energia ou se mais do que o número suportado de dispositivos falharem em uma configuração de ZFS redundante. Essa versão oferece novos recursos de comandos para recuperar seu conjunto de armazenamento danificado. No entanto, utilizar esse recurso de recuperação significa que as últimas transações que ocorreram previamente à interrupção do conjunto podem ser perdidas.

Tanto o comando `zpool clear` quanto o `zpool import` suportam a opção `-F` para possivelmente recuperar um conjunto danificado. Além disso, executar os comandos `zpool status`, `zpool clear` ou `zpool import` reporta automaticamente um conjunto danificado e esses comandos descrevem como recuperá-lo.

Para mais informações, consulte [“Reparando o dano de todo o pool de armazenamento do ZFS”](#) na página 313.

Aprimoramentos no dispositivo de log do ZFS

Versão 10 9/10 do Oracle Solaris: estão disponíveis os seguintes aprimoramentos de dispositivo de log:

- A propriedade `logbias`: é possível utilizar essa propriedade para oferecer uma dica ao ZFS sobre a manipulação de solicitações síncronas para um conjunto de dados específico. Se `logbias` estiver definida como `latência`, o ZFS utiliza os dispositivos de log separados do conjunto, se houver algum, para manipular as solicitações em baixa latência. Se `logbias` estiver definida como `throughput`, o ZFS não utiliza os dispositivos de log separados do conjunto. Em vez disso, o ZFS otimiza operações síncronas para `throughput` de conjunto global e uso eficiente de recursos. O valor padrão é `latência`. Para a maioria das configurações, é recomendado o valor padrão. Utilizar o valor `logbias=throughput` pode melhorar o desempenho da gravação de arquivos de banco de dados.
- Remoção do dispositivo de log: agora é possível remover um dispositivo de log de um conjunto de armazenamento do ZFS utilizando o comando `zpool remove`. Um dispositivo de log único pode ser removido especificando o nome do dispositivo. Um dispositivo de log espelhado pode ser removido especificando o espelho de nível superior para o log. Quando um dispositivo de log separado é removido do sistema, os registros de transação ZIL são gravados no conjunto principal.

Dispositivos virtuais de nível superior redundantes agora são identificados com um identificador numérico. Por exemplo, em um conjunto de armazenamento espelhado de dois discos, o dispositivo virtual de nível superior é `mirror-0`.

Para obter mais informações, consulte o [Exemplo 4-3](#).

RAIDZ de paridade tripla (`raidz3`)

Versão 10 9/10 do Oracle Solaris: nessa versão do Solaris, uma configuração RAID-Z redundante agora pode possuir paridade única, dupla ou tripla, o que significa que uma, duas ou três falhas de dispositivo podem ser sustentadas respectivamente, sem qualquer perda de dados. Você pode especificar a palavra-chave `raidz3` para uma configuração de RAID-Z de paridade tripla. Para mais informações, consulte [“Criando um conjunto de armazenamento RAID-Z” na página 74](#).

Mantendo instantâneos do ZFS

Versão 10 9/10 do Oracle Solaris: se implementar diferentes políticas de instantâneo automáticas de maneira que instantâneos mais antigos estejam sendo inadvertidamente destruídos pelo `zfs receive` por não existirem mais no lado de envio, você pode considerar a utilização do recurso de contenção de instantâneos nessa versão do Solaris.

Manter um instantâneo o impede de ser destruído. Além disso, esse recurso permite que um instantâneo com clones seja excluído, ficando pendente a remoção do último clone utilizando o comando `zfs destroy -d`.

É possível manter um instantâneo ou um conjunto de instantâneos. Por exemplo, a sintaxe a seguir coloca uma marcação de contenção `keep` em `tank/home/cindys/snap@1`.

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

Para mais informações, consulte [“Contendo instantâneos do ZFS” na página 223](#).

Aprimoramentos de substituição de dispositivo do ZFS

Versão 10 9/10 do Oracle Solaris: nessa versão do Solaris, um evento de sistema ou `sysevent` é fornecido quando um dispositivo subjacente é expandido. O ZFS foi aprimorado para reconhecer esses eventos e ajustar o conjunto com base no novo tamanho do LUN expandido, dependendo da configuração da propriedade `autoexpand`. É possível utilizar a propriedade do conjunto `autoexpand` para ativar ou desativar a expansão automática do conjunto quando um evento de expansão de LUN dinâmico for recebido.

Esses recursos permitem que você expanda um LUN e o conjunto resultante pode acessar o espaço expandido sem precisar exportar e importar o conjunto ou reinicializar o sistema.

Por exemplo, a expansão LUN automática é ativada no conjunto `tank`.

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

Ou você pode criar o conjunto com a propriedade `autoexpand` ativada.

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

A propriedade `autoexpand` está desativada por padrão para que você possa decidir se deseja ou não a expansão do LUN.

Um LUN também pode ser expandido utilizando o comando `zpool online -e`. Por exemplo:

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

É possível redefinir a propriedade `autoexpand` depois do LUN ser anexado ou disponibilizado com a utilização do recurso `zpool replace`. Por exemplo, o conjunto a seguir é criado com um disco de 8 GB (`c0t0d0`). O disco de 8 GB é substituído por um disco de 16 GB (`c1t13d0`), mas o tamanho do conjunto não é expandido até que a propriedade `autoexpand` seja ativada.

```
# zpool create pool c0t0d0
# zpool list
NAME    SIZE  ALLOC  FREE   CAP  HEALTH  ALROOT
```

```

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

```

Outra maneira de expandir o LUN no exemplo acima sem ativar a propriedade `autoexpand` é utilizar o comando `zpool online -e` mesmo que o dispositivo já esteja online. Por exemplo:

```

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

```

Os aprimoramentos adicionais de substituição de dispositivo nessa versão incluem os recursos a seguir:

- Em versões anteriores, o ZFS não era capaz de substituir um disco existente por outro disco ou anexar um disco se o disco de substituição fosse de um tamanho ligeiramente diferente. Nessa versão, é possível substituir um disco existente por outro disco ou anexar um novo disco que seja nominalmente do mesmo tamanho, desde que o conjunto não esteja cheio.
- Nessa versão, não é necessário reinicializar o sistema ou exportar e importar um conjunto para expandir um LUN. Conforme descrito acima, você pode ativar a propriedade `autoexpand` ou utilizar o comando `zpool online -e` para expandir o tamanho total de um LUN.

Para obter informações sobre a substituição de dispositivos, consulte [“Substituindo dispositivos em um pool de armazenamento”](#) na página 95.

Suporte de instalação do ZFS e Flash

Versão 10 10/09 do Solaris: nessa versão do Solaris, é possível configurar um perfil JumpStart para identificar um arquivo Flash de um conjunto raiz ZFS. Para mais informações, consulte [“Instalando um sistema de arquivos raiz ZFS \(instalação de arquivo do Oracle Solaris Flash\)”](#) na página 137.

Cotas de usuário e grupo do ZFS

Versão 10 10/09 do Solaris: em versões anteriores do Solaris, você podia aplicar cotas e reservas para um sistema de arquivos ZFS para gerenciar e reservar espaço em disco.

Nessa versão do Solaris, é possível definir uma cota na quantidade de espaço em disco utilizado por arquivos que pertencem a um determinado usuário ou grupo. Você pode considerar definir cotas de usuário e grupo em um ambiente com um grande número de usuários ou grupos.

É possível definir uma cota de usuário utilizando a propriedade `zfs userquota`. Para definir uma cota de grupo, utilize a propriedade `zfs groupquota`. Por exemplo:

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

Você pode exibir a cota atual de usuários ou grupos configurando como a seguir:

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

Exiba informações de cota gerais como a seguir:

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

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

É possível exibir a utilização de espaço de um usuário individual visualizando a propriedade `userused@usuário`. A utilização do espaço em disco de um grupo pode ser visualizada através da utilização da propriedade `groupused@grupo`. Por exemplo:

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

Para mais informações sobre como definir cotas de usuário, consulte [“Definindo cotas e reservas do ZFS” na página 215](#).

Herança ACL de passar para permissões de executar ZFS

Versão 10/10/09 do Solaris: em versões anteriores do Solaris, você podia aplicar a herança ACL para que todos os arquivos fossem criados com as permissões `0664` ou `0666`. Nessa versão, se você quiser opcionalmente incluir o bit executar do modo de criação de arquivo no ACL herdado, é possível definir o modo `aclinherit` para passar a permissão de executar ao ACL herdado.

Se `aclinherit=passthrough-x` estiver ativado em um conjunto de dados ZFS, você pode incluir permissões de executar para um arquivo de saída que seja gerado a partir das ferramentas de compilação `cc` ou `gcc`. Se a ACL herdada não inclui permissões de executar, então a saída executável do compilador não será executável até que você utilize o comando `chmod` para alterar as permissões do arquivo.

Para obter mais informações, consulte o [Exemplo 8–12](#).

Aprimoramentos de propriedades do ZFS

Solaris 10 10/09 e Oracle Solaris 10 9/10: os aprimoramentos do sistema de arquivos do ZFS a seguir estão incluídos nessas versões.

- **Aprimoramentos de propriedade de fluxo de instantâneos do ZFS:** é possível definir uma propriedade recebida que seja diferente de sua configuração de propriedade local. Por exemplo, é possível receber um fluxo com a propriedade de compactação desativada, mas você deseja a compactação ativada no sistema de arquivos receptor. Isso significa que o fluxo recebido possui um valor de compactação recebido de `off` e um valor de compactação local `deon`. Uma vez que o valor local ignora o valor recebido, não é necessário se preocupar com a configuração no lado de envio substituindo o valor do lado recebido. O comando `zfs get` mostra o valor efetivo da propriedade de compactação na coluna `VALUE`.

As novas propriedades e opções de comando do ZFS para suportar valores de propriedade locais e de envio são como a seguir:

- Utilize o `zfs inherit -S` para reverter um valor de propriedade local para o valor recebido, se houver algum. Se uma propriedade não possui um valor recebido, o comportamento do comando `zfs inherit -S` é o mesmo do comando `zfs inherit` sem a opção `-S`. Se a propriedade não possui um valor recebido, o comando `zfs inherit` mascara o valor recebido com o valor herdado até que a emissão de um comando `zfs inherit -S` o reverta para o valor recebido.
- É possível utilizar o `zfs get -o` para incluir a nova coluna `RECEIVED` não padrão. Ou utilize o comando `zfs get -o all` para incluir todas as colunas, inclusive a `RECEIVED`.
- É possível utilizar a opção `zfs send -p` para incluir propriedades no fluxo de envio sem a opção `-R`.

Além disso, é possível utilizar a opção `zfs send -e` para utilizar o último elemento do nome do instantâneo enviado para determinar o novo nome do instantâneo. O exemplo a seguir envia o instantâneo `poola/bee/cee@1` ao sistema de arquivos `poold/eee` e utiliza somente o último elemento (`cee@1`) do nome do instantâneo para criar o instantâneo e o sistema de arquivos recebido.

```
# 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    -
```

- **Definindo as propriedades do sistema de arquivos do ZFS no momento da criação do conjunto:** é possível definir as propriedades do sistema de arquivos do ZFS quando o conjunto é criado. No exemplo a seguir, a compactação é ativada no sistema de arquivos do ZFS que é criado quando o conjunto é criado:

```
# zpool create -O compression=on pool mirror c0t1d0 c0t2d0
```

- **Definindo as propriedades do cache em um sistema de arquivos do ZFS:** duas novas propriedades do sistema de arquivos do ZFS permitem que você controle o que é armazenado em cache no cache primário (ARC) e no cache secundário (L2ARC). As propriedades do cache são definidas como segue:
 - `primarycache` – Controla o que é colocado no cache no ARC.
 - `secondarycache` – Controla o que é colocado no cache no L2ARC.
 - Os valores possíveis para ambas as propriedades – `all`, `none` e `metadata`. Se forem definidas como `todos`, os dados do usuário e os metadados serão armazenados em cache. Se forem definidas como `nenhum`, nem os dados do usuário nem os metadados serão armazenados em cache. Se forem definidas como `metadados`, somente os metadados serão armazenados em cache. O padrão é `all`.

É possível definir essas propriedades em um sistema de arquivos existente ou quando um sistema de arquivos é criado. Por exemplo:

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

Quando essas propriedades são definidas no sistema de arquivos existente, somente E/S nova é armazenada em cache com base no valor dessas propriedades.

Alguns ambientes de banco de dados podem se beneficiar de não colocar os dados do usuário no cache. É necessário determinar se as definições das propriedades do cache são apropriadas para seu ambiente.

- **Visualizar propriedades de contabilidade de espaço em disco:** novas propriedades de sistema de arquivos de somente leitura ajudam a identificar o uso do espaço em disco para clones, sistemas de arquivos, volumes e instantâneos. As propriedades são as seguintes:
 - `usedbychildren`: identifica a quantidade de espaço em disco utilizado pelos filhos desse conjunto de dados, que seria liberado se todos os filhos do conjunto de dados fossem destruídos. A abreviatura da propriedade é `usedchildren`.
 - `usedbydataset`: identifica a quantidade de espaço em disco utilizado por esse próprio conjunto de dados, que seria liberado se o conjunto de dados fosse destruído, depois de primeiramente destruir quaisquer instantâneos e remover quaisquer reservas. A abreviatura da propriedade é `usedds`.
 - `usedbyreservation`: identifica a quantidade de espaço em disco utilizado por uma reserva definida nesse conjunto de dados, que seria liberado se a reserva fosse removida. A abreviatura da propriedade é `usedreservation`.
 - `usedbysnapshots`: identifica a quantidade de espaço em disco utilizado por instantâneos desse conjunto de dados, que seria liberado se todos os instantâneos do conjunto de dados fossem destruídos. Observe que essa não é a soma das propriedades utilizadas do instantâneo, porque o espaço em disco pode ser compartilhado por instantâneos múltiplos. A abreviatura da propriedade é `usedsnap`.

Essas novas propriedades discriminam o valor da propriedade utilizado nos diversos elementos que consomem espaço em disco. Em particular, o valor da propriedade utilizado é discriminado como a seguir:

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

É possível visualizar essas propriedades ao utilizar o comando `zfs list -o space`. Por exemplo:

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

O comando acima é equivalente ao comando `zfs list`

```
-o name,avail,used,usedsnap,usedds,usedreservation,usedchild -t filesystem,volume.
```

- **Listando instantâneos:** a propriedade `listsnapshots` do conjunto controla se a exibição das informações do instantâneo estão exibidos pelo comando `zfs list`. O valor padrão é `on`, o que significa que as informações do instantâneo são exibidas por padrão.

Se seu sistema possuir muitos instantâneos do ZFS e você desejar desativar a exibição das informações do instantâneo no comando `zfs list`, desabilite o `listsnapshots` devidamente como a seguir:

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

Se você desativar a propriedade `listsnapshots`, é possível utilizar o comando `zfs list -t snapshots` para listar as informações do instantâneo. Por exemplo:

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

Recuperação do dispositivo de registro do ZFS

Versão 10 10/09 do Solaris: nessa versão, o ZFS identifica as falhas de log intencionais na saída do comando `zpool status`. A Fault Management Architecture (FMA) também reporta esses erros. O ZFS e o FMA descrevem como recuperar uma falha de log intencional.

Por exemplo, se o sistema for desligado abruptamente antes que as operações de gravação síncrona sejam validadas para um conjunto com um dispositivo de log separado, você verá mensagens semelhantes as seguintes:

```
# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
        Waiting for administrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
        or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:
```

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

É possível solucionar a falha do dispositivo de log das seguintes maneiras:

- Substitua ou recupere o dispositivo de registro. Neste exemplo, o dispositivo é `c0t5d0`.
- Coloque o dispositivo de registro de volta on-line.

```
# zpool online pool c0t5d0
```

- Redefina a condição de erro do dispositivo de log falho.

```
# zpool clear pool
```

Se desejar recuperar esse erro sem substituir o dispositivo de log falho, é possível limpar o erro com o comando `zpool clear`. Nesse cenário, o conjunto irá operar em um modo degradado e os logs serão gravados no conjunto principal até que o dispositivo de log separado seja substituído.

Considere o uso de dispositivos de log espelhados para evitar o cenário de falha do dispositivo de log.

Utilizando dispositivos de cache no conjunto de armazenamento do ZFS

Versão 10 10/09 do Solaris: nessa versão, ao criar um conjunto é possível especificar *dispositivos de cache*, que são utilizados para armazenar em cache dados do conjunto de armazenamento.

Os dispositivos de cache fornecem uma camada adicional de cache entre a memória principal e o disco. A utilização dos dispositivos de cache oferece a maior melhoria de desempenho para cargas de trabalho de leitura aleatória do conteúdo geralmente estático.

Um ou mais dispositivos de cache podem ser especificados quando se cria o conjunto. Por exemplo:

```
# zpool create pool mirror c0t2d0 c0t4d0 cache c0t0d0
# zpool status pool
pool: pool
state: ONLINE
scrub: none requested
config:
```

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

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

Depois que os dispositivos de cache são adicionados, eles gradualmente são preenchidos com o conteúdo da memória principal. Dependendo do tamanho do dispositivo de cache, poderá demorar mais de uma hora para que os dispositivos sejam preenchidos. A capacidade e as leituras podem ser monitoradas usando o comando `zpool iostat` como segue:

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

Os dispositivos de cache podem ser adicionados ou removidos de um conjunto depois dele ser criado.

Para mais informações, consulte “Criando um conjunto de armazenamento do ZFS com dispositivos de cache” na página 76 e Exemplo 4–4.

Migração de região em um ambiente do ZFS

Versão 10 5/09 do Solaris: essa versão amplia o suporte à migração de regiões em um ambiente do ZFS com o Oracle Solaris Live Upgrade. Para obter mais informações, consulte “Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (no mínimo Solaris 10 5/09)” na página 155.

Para obter uma lista dos problemas conhecidos dessa versão, consulte as notas de versão do Solaris 10 5/09.

Suporte à inicialização e instalação do ZFS

Versão 10 10/08 do Solaris 10: essa versão permite que você instale e inicialize um sistema de arquivos raiz do ZFS. Você pode usar a opção de instalação inicial ou o recurso JumpStart para instalar um sistema de arquivos raiz ZFS. Ou você pode utilizar o recurso Oracle Solaris Live Upgrade para migrar um sistema de arquivos raiz do UFS para um sistema de arquivos raiz do ZFS. O suporte do ZFS para dispositivos de permuta e despejo também é oferecido. Para mais informações, consulte Capítulo 5, “Instalando e inicializando um sistema de arquivos raiz ZFS do Oracle Solaris”.

Para obter uma lista de problemas conhecidos dessa versão, visite o site a seguir:

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

Consulte também as notas de versão do Solaris 10 10/08.

Revertendo um conjunto de dados sem desmontar

Versão 10 10/08 do Solaris: essa versão permite que você reverta um conjunto de dados sem desmontá-lo primeiro. Dessa forma, a opção `zfs rollback - f` não é mais necessária para forçar uma operação de desmontagem. A opção `- f` não possui mais suporte e é ignorada se for especificada.

Aprimoramentos no comando `zfs send`

Versão 10 10/08 do Solaris: essa versão inclui os aprimoramentos a seguir no comando `zfs send`. Ao utilizar esse comando, é possível efetuar as tarefas a seguir:

- Enviar todos os fluxos incrementais de um instantâneo para um instantâneo cumulativo. Por exemplo:

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

Essa sintaxe envia todos os instantâneos incrementais entre fs@snapA e fs@snapD para fs@combo.

- Enviar um fluxo incremental de um instantâneo original para criar um clone. O instantâneo original já deve existir no lado de recepção para aceitar o fluxo incremental. Por exemplo:

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

- Enviar um fluxo de replicação de todos os sistemas de arquivos descendentes até os instantâneos com nome. Quando recebidos, todas as propriedades, instantâneos, sistemas de arquivo descendentes e clones são preservados. Por exemplo:

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

Para um exemplo mais amplo, consulte o [Exemplo 7-1](#).

- Enviar um fluxo de replicação incremental. Por exemplo:

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

Para um exemplo mais amplo, consulte o [Exemplo 7-1](#).

Para mais informações, consulte “[Enviando e recebendo fluxos complexos de instantâneos do ZFS](#)” na página 233.

Cotas e reservas do ZFS somente para dados do sistema de arquivos

Versão 10 10/08 do Solaris: além dos recursos de reserva e cota do ZFS, essa versão inclui as cotas e reservas do conjunto de dados que não incluem descendentes, tais como instantâneos e clones, na contabilidade do espaço em disco.

- A propriedade `refquota` impõe um limite rígido sobre a quantidade de espaço em disco que um conjunto de dados pode utilizar. Esse limite rígido não inclui o espaço em disco utilizado pelos descendentes, como instantâneos e clones.
- A propriedade `refreservation` define a quantidade mínima de espaço em disco que é garantida para um conjunto de dados, não incluindo seus descendentes.

Por exemplo, é possível definir um limite ref cota de 10 GB para `studentA` que define um limite rígido de 10 GB do espaço em disco *referenciado*. Para obter flexibilidade adicional, você pode ajustar uma cota de 20 GB que lhe permite gerenciar os instantâneos do `studentA`.

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

Para mais informações, consulte [“Definindo cotas e reservas do ZFS” na página 215](#).

Propriedades do pool de armazenamento do ZFS

Versão 10 10/08 do Solaris: as propriedades do conjunto de armazenamento do ZFS foram introduzidas em uma versão anterior. Essa versão oferece duas propriedades `cache file` e `failmode`.

A seguir, são descritas as novas propriedades do conjunto de armazenamento dessa versão:

- A propriedade `cache file`: essa propriedade controla onde as informações de configuração do conjunto são armazenadas em cache. Todos os conjuntos no cache são automaticamente importados quando o sistema é inicializado. No entanto, os ambientes de instalação e de cluster podem precisar armazenar em cache essas informações em um local diferente, de forma que os conjuntos não sejam automaticamente importados.

É possível configurar essa propriedade para armazenar em cache a configuração de conjunto em um local diferente que possa ser importado depois, utilizando o comando `zpool import -c`. Na maioria das configurações ZFS, essa propriedade não seria usada.

A propriedade `cache file` não é persistente, não sendo armazenada em disco. Essa propriedade substitui a propriedade `temporary`, que foi usada para indicar que as informações de pool não deveriam ser armazenadas em cache nas versões anteriores do Solaris.

- A propriedade `failmode`: essa propriedade determina o comportamento de uma falha catastrófica de conjunto devido à perda da conectividade do dispositivo ou à falha de todos os dispositivos no conjunto. A propriedade `failmode` pode ser configurada com estes valores: `wait`, `continue` ou `panic`. O valor padrão é `espera`, o que significa que você deve reconectar o dispositivo ou substituir um dispositivo com falha e, em seguida, limpar o erro com o comando `zpool clear`.

A propriedade `failmode` é definida como outras propriedades ZFS configuráveis, que podem ser configuradas antes ou depois do conjunto ser criado. Por exemplo:

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

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

Para obter uma descrição dessas propriedades, consulte [Tabela 4-1](#).

Aprimoramentos no histórico do comando do ZFS (zpool history)

Versão 10 10/08 do Solaris: o comando `zpool history` foi aprimorado para oferecer os novos recursos a seguir:

- As informações dos eventos do sistema de arquivos do ZFS são exibidos. Por exemplo:

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

- A opção `-l` pode ser utilizada para exibir um formato longo que inclui o nome de usuário, o nome do host e a região na qual a operação foi efetuada. Por exemplo:

```
# zpool history -l rpool
History for 'tank':
2010-06-24.13:07:58 zpool create tank mirror c2t2d0 c2t5d0 [user root on neo:global]
2010-06-24.13:08:23 zpool scrub tank [user root on neo:global]
2010-06-24.13:38:42 zpool clear tank [user root on neo:global]
2010-06-29.11:44:18 zfs create tank/home [user root on neo:global]
2010-06-29.13:28:51 zpool clear tank c2t5d0 [user root on neo:global]
2010-06-30.14:07:40 zpool add tank spare c2t1d0 [user root on neo:global]
```

- A opção `-i` pode ser utilizada para exibir informações de eventos internos para fins de diagnóstico. Por exemplo:

```
# zpool history -i tank
History for 'tank':
2010-06-24.13:07:58 zpool create tank mirror c2t2d0 c2t5d0
2010-06-24.13:08:23 [internal pool scrub txg:6] func=1 mintxg=0 maxtxg=6
2010-06-24.13:08:23 [internal pool create txg:6] pool spa 22; zfs spa 22; zpl 4; uts neo 5.10 Generic_142909-13 sun4u
2010-06-24.13:08:23 [internal pool scrub done txg:6] complete=1
2010-06-24.13:08:23 zpool scrub tank
2010-06-24.13:38:42 zpool clear tank
2010-06-24.13:38:42 [internal pool scrub txg:69] func=1 mintxg=3 maxtxg=8
2010-06-24.13:38:42 [internal pool scrub done txg:69] complete=1
2010-06-29.11:44:18 [internal create txg:14241] dataset = 34
2010-06-29.11:44:18 zfs create tank/home
2010-06-29.13:28:51 zpool clear tank c2t5d0
2010-06-30.14:07:40 zpool add tank spare c2t1d0
```

Para mais informações sobre o uso do comando `zpool history`, consulte [“Resolvendo problemas com o ZFS” na página 293](#).

Atualizando os sistemas de arquivos do ZFS (zfs upgrade)

Versão 10 10/08 do Solaris: o comando `zfs upgrade` foi incluído nessa versão para proporcionar futuros aprimoramentos do sistema de arquivos do ZFS aos sistemas de arquivos existentes. Os conjuntos de armazenamento do ZFS possuem um recurso de atualização semelhante para fornecer aprimoramentos aos conjuntos de armazenamento existentes.

Por exemplo:

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

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

Observação – Os sistemas de arquivos atualizados e quaisquer fluxos criados a partir desses sistemas de arquivos atualizados através do comando `zfs send` não estão acessíveis em sistemas que executam versões mais antigas do software.

Administração delegada do ZFS

Versão 10 10/08 do Solaris: nessa versão, é possível conceder permissões detalhadas para permitir que usuários não privilegiados efetuem tarefas de administração do ZFS.

É possível utilizar os comandos `zfs allow` e `zfs unallow` para delegar e remover permissões.

É possível modificar a administração delegada com a propriedade de delegação do conjunto. Por exemplo:

```
# 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
```

Por padrão, a propriedade `delegation` é habilitada.

Para mais informações, consulte [Capítulo 9, “Administração delegada de ZFS”](#) e [zfs\(1M\)](#).

Configurando dispositivos de log do ZFS separados

Versão 10 10/08 do Solaris: o log intencional do ZFS (ZIL) é oferecido para satisfazer as exigências de POSIX para transações síncronas. Por exemplo, em geral os bancos de dados exigem que suas transações estejam em dispositivos de armazenamento estáveis ao retornar de

uma chamada do sistema. NFS e outros aplicativos também podem utilizar `fsync()` para garantir a estabilidade dos dados. Por padrão, o ZIL é alocado a partir dos blocos no conjunto de armazenamento principal. Nessa versão do Solaris, você pode decidir se deseja que os blocos ZIL continuem a ser alocados a partir do conjunto de armazenamento principal ou a partir de um dispositivo de log separado. Pode ser possível obter melhor desempenho utilizando dispositivos de log intencional separados no conjunto de armazenamento do ZFS, como com NVRAM ou um disco dedicado.

Os dispositivos de log intencional do ZFS não estão relacionados aos arquivos de log do banco de dados.

É possível configurar o dispositivo de log do ZFS quando o conjunto de armazenamento for criado ou depois da sua criação. Para exemplos sobre a configuração dos dispositivos de log, consulte “[Criando um conjunto de armazenamento do ZFS com dispositivos de log](#)” na página 75 e “[Adicionando dispositivos a um pool de armazenamento](#)” na página 82.

É possível anexar um dispositivo de log a um dispositivo de log existente para criar um dispositivo de log espelhado. Essa operação é idêntica à anexação de um dispositivo em um conjunto de armazenamento não espelhado.

Ao determinar se a configuração de um dispositivo de registro ZFS é apropriada para o seu ambiente, leve em consideração o seguinte:

- Qualquer melhoria no desempenho visto pela implementação de um dispositivo de log separado depende do tipo de dispositivo, da configuração de hardware do conjunto e da carga de trabalho do aplicativo. Para obter informações preliminares sobre desempenho, consulte este blog:
http://blogs.sun.com/perrin/entry/slog_blog_or_logging_on
- Os dispositivos de log podem ser não replicado ou espelhados, mas o RAID-Z não é suportado por dispositivos de log.
- Se um dispositivo de log separado não for espelhado e o dispositivo que contém o log falhar, o armazenamento dos blocos de log é revertido ao conjunto de armazenamento.
- Os dispositivos de log podem ser adicionados, substituídos, anexados, desanexados, importados e exportados como parte do conjunto de armazenamento maior. Dispositivos de log podem ser removidos a partir da versão 10 9/10 do Solaris.
- O tamanho mínimo de um dispositivo de log é o mesmo do tamanho mínimo de cada dispositivo em um conjunto, que é de 64 MB. A quantidade de dados em atividade que pode ser armazenada em um dispositivo de registro é relativamente pequena. Os blocos de registro são liberados quando a transação de registro (chamada do sistema) é enviada.
- O tamanho máximo de um dispositivo de log deve ser aproximadamente a metade do tamanho da memória física, porque essa é a quantidade máxima de possíveis dados em atividade que podem ser armazenados. Por exemplo, se um sistema possuir 16 GB de memória física, considere que o tamanho máximo de um dispositivo de log será de 8 GB.

Criando conjuntos de dados do ZFS intermediários

Versão 10 10/08 do Solaris: é possível utilizar a opção `-p` com os comandos `zfs create`, `zfs clone` e `zfs rename` para criar rapidamente um conjunto de dados intermediário não existente, se ele ainda não existir.

No exemplo a seguir, conjuntos de dados do ZFS (`users/area51`) são criados no conjunto de armazenamento `datab`.

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

O conjunto de dados intermediário já existe durante a operação de criação, a operação será concluída com êxito.

As propriedades especificadas se aplicam ao conjunto de dados de destino e não ao conjunto de dados intermediário. Por exemplo:

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

O conjunto de dados intermediário é criado com o ponto padrão de montagem. Quaisquer propriedades adicionais são desativadas para o conjunto de dados intermediário. Por exemplo:

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

Para obter mais informações, consulte [zfs\(1M\)](#).

Aprimoramentos de conexão automática do ZFS

Versão 10 10/08 do Solaris: nessa versão, o ZFS responde de maneira mais eficiente aos dispositivos removidos e pode agora identificar automaticamente dispositivos que são inseridos.

- Você pode substituir um dispositivo atual por um dispositivo equivalente, sem precisar usar o comando `zpool replace`.

A propriedade `autoreplace` controla a substituição automática de dispositivos. Se for definido como `off`, a substituição do dispositivo deve ser iniciada pelo administrador utilizando o comando `zpool replace`. Se for definido como `on`, qualquer dispositivo novo encontrado no mesmo local físico de um dispositivo que antes pertencia ao conjunto será automaticamente formatado e substituído. O comportamento padrão é `off`.

- O estado do conjunto de armazenamento REMOVED é fornecido quando um dispositivo ou de reserva tiver sido fisicamente removido enquanto o sistema estava em execução. Um dispositivo de reserva é substituído pelo dispositivo removido, se disponível.
- Se um dispositivo for removido e depois inserido, o dispositivo será colocado on-line. Se um de reserva foi ativado quando o dispositivo foi reinserido, o de reserva é removido quando a operação on-line for concluída.
- Quando os dispositivos são removidos ou inseridos, a detecção automática depende do hardware e pode não ser suportada em todas as plataformas. Por exemplo, os dispositivos USB são configurados automaticamente ao serem inseridos. No entanto, você pode ter que usar o comando `cfgadm -c configure` para configurar uma unidade SATA.
- Os de reservas são verificados periodicamente para garantir que estejam on-line e disponíveis.

Para mais informações, consulte [zpool\(1M\)](#).

Renomear recursivamente instantâneos do ZFS (`zfs rename -r`)

Versão 10 10/08 do Solaris: é possível renomear recursivamente todos os instantâneos descendentes do ZFS utilizando o comando `zfs rename -r`. Por exemplo:

Primeiro, um instantâneo de um conjunto de sistemas de arquivos do ZFS é criado.

```
# zfs snapshot -r users/home@today
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
users                               216K  16.5G  20K    /users
users/home                          76K   16.5G  22K    /users/home
users/home@today                    0     -      22K    -
users/home/markm                   18K   16.5G  18K    /users/home/markm
users/home/markm@today              0     -      18K    -
users/home/marks                   18K   16.5G  18K    /users/home/marks
users/home/marks@today              0     -      18K    -
users/home/neil                    18K   16.5G  18K    /users/home/neil
users/home/neil@today               0     -      18K    -
```

Depois, os instantâneos são renomeados no dia seguinte.

```
# zfs rename -r users/home@today @yesterday
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
users                               216K  16.5G  20K    /users
users/home                          76K   16.5G  22K    /users/home
users/home@yesterday                0     -      22K    -
users/home/markm                   18K   16.5G  18K    /users/home/markm
users/home/markm@yesterday          0     -      18K    -
users/home/marks                   18K   16.5G  18K    /users/home/marks
users/home/marks@yesterday          0     -      18K    -
```

```
users/home/neil          18K  16.5G   18K  /users/home/neil
users/home/neil@yesterday 0      -    18K  -
```

Um instantâneo é o único tipo de conjunto de dados que pode ser renomeado recursivamente.

Para mais informações sobre instantâneos, consulte “[Visão geral dos instantâneos do ZFS](#)” na página 221 e esta entrada do blog que descreve como criar instantâneos sucessivos:

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

gzip A compactação está disponível para ZFS

Versão 10 10/08 do Solaris: nessa versão do Solaris, é possível definir a compactação gzip em sistemas de arquivos do ZFS além da compactação lzjb. É possível especificar a compactação como gzip ou gzip-*N*, em que *N* é de 1 a 9. Por exemplo:

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

Para mais informações sobre a configuração das propriedades do ZFS, consulte “[Definindo propriedades do ZFS](#)” na página 203.

Armazenamento de várias cópias de dados de usuário do ZFS

Versão 10 10/08 do Solaris: como um recurso de segurança, os metadados do sistema de arquivos do ZFS são armazenados automaticamente várias vezes em diferentes discos, se possível. Este recurso é conhecido como *blocos idem*.

Nessa versão do Solaris, também é possível armazenar cópias múltiplas dos dados do usuário que também são armazenadas por sistema de arquivos ao utilizar o comando `zfs set copies`. Por exemplo:

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

Os valores disponíveis são 1, 2 ou 3. O valor padrão é 1. Essas cópias estão em acréscimo a qualquer redundância de nível de conjunto, tal como na configuração espelhada ou RAID-Z.

Os benefícios do armazenamento de cópias múltiplas dos dados do usuário do ZFS são os seguintes:

- Melhora a retenção de dados, permitindo recuperações de falhas de leituras em blocos irrecuperáveis, como falhas de mídia (comumente conhecidas como *bit rot*) em todas as configurações do ZFS.
- Oferece proteção de dados, mesmo quando apenas um disco único está disponível.
- Permite que você selecione as políticas de proteção de dados por bases de sistema de arquivos, além das capacidades do conjunto de armazenamento.

Observação – Dependendo da alocação de blocos idênticos no conjunto de armazenamento, as cópias múltiplas podem ser colocadas em um único disco. Uma falha de disco cheio subsequente pode fazer com que todos os blocos idênticos fiquem indisponíveis.

É possível levar em consideração o uso de blocos idênticos ao criar acidentalmente um conjunto não redundante e quando precisar definir as políticas de retenção de dados.

Para obter uma descrição detalhada de como armazenar cópias múltiplas em um sistema com um conjunto de disco único ou um conjunto de disco múltiplo pode afetar a proteção geral dos dados, consulte este blog:

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

Para mais informações sobre a configuração das propriedades do ZFS, consulte “[Definindo propriedades do ZFS](#)” na página 203.

Saída `zpool status` aprimorada

Versão 10 8/07 do Solaris: é possível utilizar o comando `zpool status -v` para exibir uma lista de arquivos com erros persistentes. Anteriormente, era necessário utilizar o comando `find -inum` para identificar os nomes de arquivos na lista de inodes exibida.

Para mais informações sobre a exibição de uma lista de arquivos com erros persistentes, consulte “[Reparando arquivos ou diretórios corrompidos](#)” na página 312.

Aprimoramentos na iSCSI do Solaris e ZFS

Versão 10 8/07 do Solaris: nessa versão do Solaris, é possível criar um volume ZFS como um dispositivo de destino do iSCSI do Solaris definindo a propriedade `shareiscsi` no volume ZFS. Esse método é uma forma conveniente de configurar rapidamente um destino iSCSI do Solaris. Por exemplo:

```
# zfs create -V 2g tank/volumes/v2
# zfs set shareiscsi=on tank/volumes/v2
# iscsitadm list target
```

```
Target: tank/volumes/v2
  iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
  Connections: 0
```

Depois da criação do destino iSCSI, é possível configurar o iniciador iSCSI. Para mais informações sobre a configuração do iniciador iSCSI do Solaris, consulte [Capítulo 14, “Configuring Oracle Solaris iSCSI Targets and Initiators \(Tasks\)”](#), no *System Administration Guide: Devices and File Systems*.

Para mais informações sobre gerenciamento de volumes do ZFS como um destino de iSCSI, consulte [“Usando um volume ZFS como um destino iSCSI Solaris”](#) na página 279.

Histórico do comando ZFS (zpool history)

Versão 10 8/07 do Solaris: nessa versão do Solaris, o ZFS registra automaticamente comandos `zfs` e `zpool` bem-sucedidos que modificam as informações de estado do conjunto. Por exemplo:

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

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

Esses recursos permitem que você ou a equipe de suporte do Oracle identifique os comandos *reais* do ZFS que foram executados para corrigir um cenário de erro.

É possível identificar um conjunto de armazenamento específico com o comando `zpool history`. Por exemplo:

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

Nessa versão do Solaris, o comando `zpool history` não registra o *ID de usuário*, o *nome do host* nem o *nome da região*. No entanto, essas informações são gravadas a partir da versão 10 10/08 do Solaris. Para mais informações, consulte [“Aprimoramentos no histórico do comando do ZFS \(zpool history\)”](#) na página 32.

Para mais informações sobre soluções de problemas do ZFS, consulte [“Resolvendo problemas com o ZFS”](#) na página 293.

Melhorias das propriedades do ZFS

Propriedade `xattr` do ZFS

Versão 10 8/07 do Solaris: é possível utilizar a propriedade `xattr` para desativar ou ativar os atributos estendidos para um sistema de arquivos do ZFS específico. O valor padrão é `on`. Para obter uma descrição das propriedades do ZFS, consulte [“Introduzindo as propriedades do ZFS” na página 187](#).

Propriedade `canmount` do ZFS

Versão 10 8/07 do Solaris: a nova propriedade `canmount` permite que você especifique se um conjunto de dados pode ser montado através da utilização do comando `zfs mount`. Para mais informações, consulte [“A propriedade `canmount`” na página 198](#).

Propriedades de usuário do ZFS

Versão 10 8/07 do Solaris: além das propriedades nativas padrão que podem ser utilizadas para exportar estatísticas internas ou controlar o comportamento do sistema de arquivos do ZFS, o ZFS fornece propriedades do usuário. As propriedades do usuário não afetam o comportamento do ZFS, mas podem ser utilizadas para anotar conjuntos de dados com informações significativas no ambiente.

Para mais informações, consulte [“Propriedades de usuário do ZFS” na página 200](#).

Definindo propriedades ao criar sistemas de arquivos do ZFS

Versão 10 8/07 do Solaris: nessa versão do Solaris, é possível definir propriedades ao criar um sistema de arquivos, não apenas depois do sistema de arquivos ter sido criado.

Os exemplos abaixo ilustram a sintaxe equivalente:

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

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

Exibindo todas as informações do sistema de arquivos do ZFS

Versão 10 8/07 do Solaris: nessa versão do Solaris, é possível utilizar várias formulários do comando `zfs get` para exibir informações sobre todos os conjuntos de dados caso você não especifique um conjunto de dados ou caso você especifique todos. Em versões anteriores, não era possível recuperar todas as informações do conjunto de dados com o comando `zfs get`.

Por exemplo:

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

Nova opção `zfs receive -F`

Versão 10 8/07 do Solaris: nessa versão do Solaris, é possível utilizar a nova opção `-F` no comando `zfs receive` para forçar uma reversão do sistema de arquivos ao instantâneo mais recente antes do recebimento ser iniciado. O uso dessa opção pode ser necessário quando o sistema de arquivos for modificado depois da reversão ocorrer, mas antes de o recebimento ser iniciado.

Para mais informações, consulte [“Recebendo um instantâneo do ZFS” na página 232](#).

Instantâneos recursivos do ZFS

Versão 10 11/06 do Solaris: ao utilizar o comando `zfs snapshot` para criar um instantâneo do sistema de arquivos, é possível utilizar a opção `-r` para criar recursivamente instantâneos para todos os sistemas de arquivos descendentes. Além disso, é possível utilizar a opção `-r` para destruir recursivamente todos os instantâneos descendentes quando um instantâneo é destruído.

Os instantâneos recursivos do ZFS são criados rapidamente como uma operação atômica. Os instantâneos são criados juntos (todos de uma vez) ou simplesmente não são criados. O benefício de tal operação é que os dados do instantâneo são sempre tirados em momento consistente, mesmo através de sistemas de arquivos descendentes.

Para mais informações, consulte [“Criando e destruindo instantâneos do ZFS” na página 222](#).

RAID-Z de paridade dupla (`raidz2`)

Versão 10 11/06 do Solaris: uma configuração RAID-Z redundante agora pode possuir configuração de paridade única ou dupla, o que significa que uma ou duas falhas de dispositivo podem ser sustentadas respectivamente, sem qualquer perda de dados. Você pode especificar a palavra-chave `raidz2` para realizar uma configuração de RAID-Z de paridade dupla. Ou você pode especificar as palavras-chave `raidz` ou `raidz1` para realizar uma configuração RAID-Z de paridade única.

Para mais informações, consulte [“Criando um conjunto de armazenamento RAID-Z” na página 74](#) ou `zpool(1M)`.

Sobressalentes para dispositivos do pool de armazenamento do ZFS

Versão 10 11/06 do Solaris: o recurso de reserva do ZFS permite que você identifique os discos que poderiam ser utilizados para substituir um dispositivo falho ou defeituoso em um ou mais conjuntos de armazenamento. Designar um dispositivo como um *de reserva* significa que, se um dispositivo ativo no conjunto falhar, o dispositivo de reserva substitui automaticamente o dispositivo falho. Ou então, é possível substituir manualmente um dispositivo em um conjunto de armazenamento por um de reserva.

Para mais informações, consulte [“Designando sobressalentes no pool de armazenamento”](#) na página 97 e `zpool(1M)`.

Substituindo um sistema de arquivos do ZFS por um clone do ZFS (`zfs promote`)

Versão 10 11/06 do Solaris: o comando `zfs promote` permite que você substitua um sistema de arquivos existente do ZFS existente por um clone desse sistema de arquivos. Esse recurso é útil quando você deseja executar testes em uma versão alternativa de um sistema de arquivos e, em seguida, tornar essa versão alternativa em um sistema de arquivos ativo.

Para mais informações, consulte [“Substituindo um sistema de arquivos ZFS por um clone do ZFS”](#) na página 229 e `zfs(1M)`.

Atualizando conjuntos de armazenamento do ZFS (`zpool upgrade`)

Versão 10 6/06 do Solaris: é possível atualizar seus conjuntos de armazenamento para uma versão mais nova a fim de obter benefícios dos recursos mais recentes por meio do comando `zpool upgrade`. Além disso, o comando `zpool status` foi modificado para notificá-lo quando os conjuntos estiverem executando versões mais antigas do ZFS.

Para mais informações, consulte [“Atualizando pools de armazenamento do ZFS”](#) na página 121 e `zpool(1M)`.

Se desejar utilizar o console de administração do ZFS em um sistema com um conjunto de uma versão anterior do Solaris, certifique-se de atualizar os conjuntos antes de utilizar o console. Para determinar se os conjuntos precisam ser atualizados, utilize o comando `zpool status`. Para obter informações sobre o console de administração do ZFS, consulte [“Gerenciamento do ZFS baseado na Web”](#) na página 44.

Comandos de restauração e backup do ZFS renomeados

Versão 10 6/06 do Solaris: nessa versão do Solaris, os comandos `zfs backup` e `zfs restore` foram renomeados como `zfs send` e `zfs receive` para uma descrição mais exata de suas funções. Esses comandos enviam e recebem representações de fluxo de dados do ZFS.

Para mais informações sobre esses comandos, consulte [“Enviando e recebendo dados do ZFS” na página 230](#).

Recuperando conjuntos de armazenamento destruídos

Versão 10 6/06 do Solaris: essa versão inclui o comando `zpool import -D`, que permite que você recupere conjuntos que foram destruído anteriormente com o comando `zpool destroy`.

Para mais informações, consulte [“Recuperando pools de armazenamento destruídos do ZFS” na página 120](#).

O ZFS é integrado com o gerenciador de falhas

Versão 10 6/06 do Solaris: essa versão inclui um mecanismo de diagnóstico do ZFS que é capaz de diagnosticar e relatar falhas do conjunto e do dispositivo. Também são relatados erros na soma de verificação, de E/S, de dispositivo e de pool associados às falhas do dispositivo e do pool.

O mecanismo de diagnóstico não inclui a análise prognóstica de erros de E/S e de soma de verificação, nem ações antecipatórias com base nas análises de falhas.

Se ocorrer uma falha no ZFS, aparecerá uma mensagem similar a seguinte:

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

Ao revisar a ação recomendada, que será seguir as direções mais específicas no comando `zpool status`, você poderá identificar e resolver rapidamente a falha.

Para obter um exemplo de recuperação de um problema do ZFS reportado, consulte [“Reparando um dispositivo faltando” na página 299](#).

O comando `zpool clear`

Versão 10 6/06 do Solaris: essa versão inclui o comando `zpool clear` para limpar contagens de erros associadas a um dispositivo ou a um conjunto. Anteriormente, as contagens de erros eram limpas quando um dispositivo em um conjunto era colocado on-line com o comando `zpool onLine`. Para mais informações, consulte “[Limpar erros de dispositivo de conjunto de armazenamento](#)” na página 95 e `zpool(1M)`.

Formato compacto da ACL NFSv4

Versão 10 6/06 do Solaris: nessa versão, é possível definir e exibir ACLs NFSv4 em dois formatos: verboso e compacto. É possível utilizar o comando `chmod` para definir qualquer um dos formatos ACL. É possível utilizar o comando `ls -V` para exibir o formato compacto da ACL. É possível utilizar o comando `ls -lv` para exibir o formato verboso da ACL.

Para mais informações, consulte “[Definindo e exibindo ACLs em arquivos ZFS no formato compacto](#)” na página 258, `chmod(1)` e `ls(1)`.

Ferramenta de monitoramento do sistema de arquivos (`fsstat`)

Versão 10 6/06 do Solaris: uma nova ferramenta de monitoração do sistema de arquivos, `fsstat`, relata as operações dos sistemas de arquivos. A atividade pode ser relatada por ponto de montagem ou por tipo de sistema de arquivos. O exemplo abaixo mostra a atividade geral do sistema de arquivos ZFS:

```
$ fsstat zfs
new name name attr attr lookup rddir read read write write
file remov chng get set ops ops ops bytes ops bytes
7.82M 5.92M 2.76M 1.02G 3.32M 5.60G 87.0M 363M 1.86T 20.9M 251G zfs
```

Para mais informações, consulte `fsstat(1M)`.

Gerenciamento do ZFS baseado na Web

Versão 10 6/06 do Solaris: uma ferramenta de gerenciamento do ZFS com base na Web, o console de administração do ZFS, permite que você efetue as tarefas administrativas a seguir:

- Criar um novo pool de armazenamento.
- Adicionar capacidade a um pool existente.
- Mover (exportar) um pool de armazenamento para outro sistema.
- Importar um pool de armazenamento exportado anteriormente para torná-lo disponível em outro sistema.

- Visualizar informações sobre pools de armazenamento.
- Criar um sistema de arquivos.
- Criar um volume.
- Criar um instantâneo de um sistema de arquivos ou um volume.
- Reverter um sistema de arquivos para um instantâneo anterior.

É possível acessar o console de administração do ZFS através de um navegador da Web seguro em:

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

Se você digitar o URL apropriado e não conseguir encontrar o console Administração do ZFS, o servidor não deve ter sido iniciado. Para iniciar o servidor, execute o seguinte comando:

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

Se quiser que o servidor seja executado automaticamente quando o sistema for inicializado, execute o seguinte comando:

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

Observação – Não é possível usar o console de gerenciamento Solaris (smc) para gerenciar pools de armazenamento e sistemas de arquivos ZFS.

O que é ZFS?

O sistema de arquivos do ZFS é um novo e revolucionário sistema de arquivos que basicamente altera a forma como os sistemas de arquivos são administrados, com recursos e benefícios que não são encontrados nos sistemas de arquivos disponíveis atualmente. O ZFS é robusto, dimensionável e fácil de administrar.

Armazenamento de ZFS em pool

O ZFS usa o conceito de *pools de armazenamento* para gerenciar armazenamentos físicos. Os sistemas de arquivos sempre foram construídos em cima de um dispositivo físico. Para abordar vários dispositivos e prevenir a redundância de dados, introduziu-se o conceito de um *gerenciador de volumes*, com o intuito de fornecer uma representação de um dispositivo único para que os sistemas de arquivos não precisassem ser modificados, a fim de aproveitar os vários dispositivos. Esse design adicionou outra camada de complexidade e, finalmente, impediu certos avanços do sistema de arquivos porque o sistema de arquivos não possuía controle sobre a localização física dos dados nos volumes virtualizados.

O ZFS elimina totalmente o gerenciamento de volumes. Em vez de forçá-lo a criar volumes virtualizados, o ZFS agrega os dispositivos em um conjunto de armazenamento. O conjunto de armazenamento descreve as características físicas do armazenamento (layout do dispositivo, redundância de dados e assim por diante) e atua como um armazenamento de dados arbitrário a partir do qual podem ser criados sistemas de arquivos. Os sistemas de arquivos já não estão mais limitados a dispositivos individuais, permitindo-lhes compartilhar espaço em disco com todos os sistemas de arquivos do conjunto. Já não é mais necessário pré-determinar o tamanho de um sistema de arquivos, visto que os sistemas de arquivos crescem automaticamente dentro do espaço em disco alocado para o conjunto de armazenamento. Quando novos armazenamentos são adicionados, todos os sistemas de arquivos que estão dentro do conjunto podem utilizar imediatamente o espaço adicional sem trabalho adicional. Em muitos aspectos, o conjunto de armazenamento trabalha de maneira similar a um sistema de memória virtual: quando uma memória DIMM é adicionada a um sistema, o sistema operacional não o força a executar comandos para configurar a memória e atribuí-la a processos individuais. Todos os processos do sistema usam automaticamente a memória adicional.

Semânticas transacionais

O ZFS é um sistema de arquivos transacional, o que significa que o estado do sistema de arquivos é sempre consistente no disco. Os sistemas de arquivos tradicionais substituem os dados no local, o que significa que, se a máquina apagar, por exemplo, entre o momento em que um bloco de dados está sendo alocado e o momento em que está sendo vinculado a um diretório, o sistema de arquivos ficará em um estado de inconsistência. Historicamente, esse problema era solucionado através da utilização do comando `fsck`. Esse comando era responsável por revisar e verificar o estado do sistema de arquivos e tentar reparar quaisquer inconsistências encontradas durante o processo. Esse problema de sistemas de arquivos inconsistentes deu muito trabalho aos administradores e nunca era garantido que o comando `fsck` resolvesse todos os possíveis problemas. Mais recentemente, os sistemas de arquivos introduziram o conceito de *ocorrências*. O processo de ocorrências registra ações em um diário diferente, que pode então ser *reproduzido* de forma segura se ocorrer uma quebra no sistema. Esse processo introduz uma sobrecarga desnecessária, pois os dados precisam ser gravados duas vezes, resultando frequentemente em novos conjuntos de problemas, tais como quando o diário não pode ser reproduzido apropriadamente.

Com um sistema de arquivos transacional, os dados são gerenciados utilizando as semânticas de *copiar ao gravar*. Os dados nunca são substituídos e nenhuma seqüência de operações é totalmente comprometida ou totalmente ignorada. Assim, o sistema de arquivos nunca será corrompido por perdas acidentais de energia ou por quebra no sistema. Embora as últimas partes de dados gravadas possam se perder, o sistema de arquivos em si será sempre consistente. Além disso, os dados síncronos (escritos com o sinalizador `O_DSYNC`) sempre serão gravados antes de retornarem, de modo que nunca se perderão.

Somas de verificação e autocorreção de dados

Com o ZFS, todos os dados e metadados são verificados ao utilizar um algoritmo de soma de verificação selecionável pelo usuário. Os sistemas de arquivos tradicionais, que oferecem a soma de verificação, efetuavam tal soma com base em blocos sem necessidade, devido à camada de gerenciamento de volumes e ao design do sistema de arquivos tradicional. O design tradicional significa que determinadas falhas, tais como gravar um bloco completo em um local incorreto, podem resultar em dados incorretos mas sem erros de soma de verificação. As somas de verificação do ZFS são armazenadas de forma que essas falhas sejam detectadas e possam ser recuperadas adequadamente. Todas as somas de verificação e recuperações de dados são efetuadas na camada do sistema de arquivos e são evidentes para os aplicativos.

Além disso, o ZFS oferece a autocorreção de dados. O ZFS suporta conjuntos de armazenamento com níveis variados de redundância de dados. Quando um bloco de dados incorreto é detectado, o ZFS busca os dados corretos em outra cópia redundante e repara os dados incorretos substituindo-os pela cópia dados corretos.

Escalabilidade não paralela

Um elemento de design fundamental do sistema de arquivos do ZFS é a escalabilidade. O sistema de arquivos em si é de 128 bits, permitindo 256 quatrilhões de zettabytes de armazenamento. Todos os metadados estão alocados dinamicamente, de forma que não há necessidade de pré-alocar inodes ou de limitar a escalabilidade do sistema de arquivos quando este for criado pela primeira vez. Todos os algoritmos foram gravados levando-se em consideração a escalabilidade. Os diretórios podem possuir até 2^{48} (256 trilhões) de entradas e não há limites para o número de sistemas de arquivos ou para o número de arquivos que pode ser contido em um sistema de arquivos.

Instantâneos do ZFS

Um *instantâneo* é uma cópia de somente leitura de um volume ou de um sistema de arquivos. Os instantâneos podem ser criados rápida e facilmente. No início, os instantâneos não utilizam espaço adicional em disco dentro do conjunto.

Conforme os dados dentro dos conjuntos de dados ativos são alterados, o instantâneo passa a utilizar mais espaço em disco devido às contínuas referências aos dados antigos. Como consequência, o instantâneo evita que os dados sejam liberados de novo no pool.

Administração simplificada

O mais importante é que o ZFS oferece um modelo de administração muito simplificado. Através do uso de um layout de sistema de arquivos hierárquico, da herança de propriedade e do gerenciamento automático de pontos de montagem e de semânticas compartilhadas de NFS,

o ZFS facilita a criação e o gerenciamento de sistemas de arquivos sem a necessidade de vários comandos ou de editar os arquivos de configuração. Você pode definir facilmente as cotas ou reservas, habilitar ou desabilitar a compactação ou gerenciar os pontos de montagem de numerosos sistemas de arquivos com um único comando. É possível examinar ou substituir dispositivos sem conhecer um conjunto separado de comandos do gerenciador de volume. É possível enviar e receber fluxos de instantâneos do sistema de arquivos.

O ZFS gerencia sistemas de arquivos através de uma hierarquia que possibilita esse gerenciamento simplificado de propriedades como cotas, reservas, compactação e pontos de montagem. Nesse modelo, sistemas de arquivos são o ponto central de controle. Os sistemas de arquivos em si são muito baratos (equivalentes à criação de um novo diretório), de modo que você é estimulado a criar um sistema de arquivos para cada usuário, projeto, área de trabalho, etc. Esse design permite que você defina pontos de gerenciamento detalhados.

Terminologia do ZFS

Esta seção descreve a terminologia básica utilizada neste livro:

ambiente de inicialização alternativo	Um ambiente de inicialização criado pelo comando <code>lucreate</code> e possivelmente atualizado pelo comando <code>luupgrade</code> , mas que não está ativo ou não é o ambiente de inicialização principal. O ambiente de inicialização alternativo pode ser alterado para o ambiente de inicialização primário por meio da execução do comando <code>luactivate</code> .
soma de verificação	Uma verificação de 256 bits dos dados em um bloco de sistemas de arquivos. A capacidade da soma de verificação pode variar do <code>fletcher4</code> (padrão) simples e rápido a verificações criptograficamente seguras como <code>SHA256</code> .
clone	Um sistema de arquivos cujo conteúdo inicial é idêntico ao conteúdo de um instantâneo. Para informações sobre clones, consulte “Visão geral dos clones do ZFS” na página 228 .
conjunto de dados	Um nome genérico dos componentes do ZFS a seguir: clones, sistemas de arquivos, instantâneos e volumes. Cada conjunto de dados é identificado por um nome único no espaço de nome do ZFS. Os conjuntos de dados são identificados com o seguinte formato:

pool/caminho [*@snapshot*]

<i>pool</i>	Identifica o nome do pool de armazenamento que contém o conjunto de dados
<i>caminho</i>	É o nome do caminho delimitado por barra para o componente do conjunto de dados
<i>instantâneo</i>	É um componente opcional que identifica um instantâneo de um conjunto de dados

Para mais informações sobre conjuntos de dados, consulte o [Capítulo 6, “Gerenciando sistemas de arquivos ZFS do Oracle Solaris”](#).

sistema de arquivos	Conjunto de dados ZFS de tipo <code>filesystem</code> montado dentro do espaço de nome padrão do sistema e que se comporta como outros sistemas de arquivos. Para mais informações sobre sistemas de arquivos, consulte o Capítulo 6, “Gerenciando sistemas de arquivos ZFS do Oracle Solaris” .
espelho	Um dispositivo virtual que armazena cópias idênticas de dados em dois ou mais discos. Se um disco falha em um espelho, qualquer outro disco desse espelho pode fornecer os mesmos dados.
pool	Grupo lógico de dispositivos que descreve o layout e as características físicas do armazenamento disponível. O espaço em disco para os conjuntos de dados é alocado a partir de um conjunto. Para mais informações sobre conjuntos de armazenamento, consulte o Capítulo 4, “Gerenciando conjuntos de armazenamento ZFS do Oracle Solaris” .
ambiente de inicialização principal	Um ambiente de inicialização que é usado pelo comando <code>lucreate</code> para construir o ambiente de inicialização alternativo. Por padrão, o ambiente de inicialização primário é o ambiente de inicialização atual. Esse padrão pode ser ignorado com o uso da opção <code>lucreate -s</code> .
RAID-Z	Um dispositivo virtual que armazena dados e paridades em discos múltiplos. Para mais informações

resilvering	<p>sobre o RAID-Z, consulte “Configuração de pool de armazenamento RAID-Z” na página 69.</p> <p>O processo de cópia de dados de um dispositivo para outro é conhecido como <i>resilvering</i>. Por exemplo, se um dispositivo do espelho é substituído ou colocado off-line, os dados de um dispositivo do espelho atualizado são copiados para o dispositivo do espelho recém-restaurado. Em produtos de gerenciamento de volumes tradicionais, faz-se referência a esse processo como <i>resincronização espelhada</i>.</p> <p>Para mais informações sobre resilvering do ZFS, consulte “Exibindo o status do resilvering” na página 309.</p>
instantâneo	<p>Uma cópia de somente de leitura de um volume ou de um sistema de arquivos em um determinado ponto do tempo.</p> <p>Para mais informações sobre instantâneos, consulte “Visão geral dos instantâneos do ZFS” na página 221.</p>
dispositivo virtual	<p>Dispositivo lógico em um pool, que pode ser um dispositivo físico, um arquivo ou um conjunto de dispositivos.</p> <p>Para mais informações sobre dispositivos virtuais, consulte “Exibindo informações do dispositivo virtual do pool de armazenamento” na página 77.</p>
volume	<p>Um conjunto de dados que representa um dispositivo de bloco. Por exemplo, você pode criar um volume ZFS como um dispositivo de permuta.</p> <p>Para obter informações sobre volumes ZFS, consulte “Volumes ZFS” na página 277.</p>

Requisitos para nomeação de componentes do ZFS

Cada componente do ZFS, tal como conjuntos de dados e conjuntos, deve ser nomeado de acordo com as regras a seguir:

- Cada componente pode conter somente caracteres alfanuméricos além dos quatro caracteres especiais a seguir:

- Sublinhado (`_`)
- Hífen (`-`)
- Dois-pontos (`:`)
- Ponto final (`.`)
- Os nomes de pool devem começar com uma letra, exceto para as seguintes restrições:
 - A seqüência inicial `c[0-9]` não é permitida.
 - O nome `log` é reservado.
 - Um nome que começa com `espelho`, `raidz`, `raidz1`, `raidz2`, `raidz3` ou `sparenão` é permitido porque esses nomes são reservados.
 - Os nomes de conjuntos não devem conter um sinal de porcentagem (`%`).
- Os nomes dos conjuntos de dados devem começar com caracteres alfanuméricos.
- Os nomes de conjunto de dados não devem conter um sinal de porcentagem (`%`).

Além disso, não são permitidos componentes vazios.

Introdução ao ZFS do Oracle Solaris

Este capítulo fornece instruções passo a passo para fazer a configuração básica do ZFS do Oracle Solaris. Até final deste capítulo, você terá uma idéia básica de como funcionam os comandos do ZFS e será capaz de criar conjuntos e sistemas de arquivos simples. Este capítulo não oferece uma visão geral abrangente e faz referência a capítulos para informações mais detalhadas do assunto tratado.

Este capítulo traz as seguintes seções:

- “Recomendações e requisitos de software e hardware do ZFS” na página 53
- “Criando um sistema de arquivos ZFS básico” na página 54
- “Criando um pool de armazenamento do ZFS” na página 55
- “Criando uma hierarquia de sistemas de arquivos ZFS” na página 56

Recomendações e requisitos de software e hardware do ZFS

Verifique as recomendações e os requisitos de hardware e software seguintes antes de tentar utilizar o software do ZFS:

- Utilize um sistema com base em SPARC ou em x86 que esteja em execução no Solaris 10 6/06 ou versão superior.
- O espaço mínimo em disco necessário para um conjunto de armazenamento é de 64 MB. O tamanho mínimo do disco é de 128 MB.
- O tamanho mínimo de memória necessária para a instalação de um sistema Solaris é de 768 MB. No entanto, para obter bom desempenho do ZFS, utilize pelo menos 1GB de memória.
- Se você criar uma configuração de disco espelhada, utilize diversos controladores.

Criando um sistema de arquivos ZFS básico

A administração do ZFS foi concebida levando-se em consideração a simplicidade. Um dos objetivos do design é reduzir o número de comandos necessários para criar um sistema de arquivos utilizável. Por exemplo, ao criar um novo conjunto, um novo sistema de arquivos do ZFS é automaticamente criado e montado.

O exemplo a seguir mostra como criar um conjunto de armazenamento espelhado básico chamado `tank` e um sistema de arquivos do ZFS chamado `tank` em um comando. Suponha que a totalidade dos discos `/dev/dsk/c2t0d0` e `/dev/dsk/c0t1d0` possa ser usada.

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

Para obter mais informações sobre configurações de pool do ZFS, consulte [“Recursos de replicação de um pool de armazenamento do ZFS” na página 69](#).

O novo sistema de arquivos do ZFS, `tank`, pode utilizar tanto espaço em disco quanto for necessário e é automaticamente montado em `/tank`.

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

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

É provável que você queira criar sistemas de arquivos adicionais dentro de um conjunto. Os sistemas de arquivos fornecem pontos de administração que permitem gerenciar diferentes conjuntos de dados dentro de um mesmo conjunto.

O exemplo a seguir mostra como criar um sistema de arquivos chamado `fs` no conjunto de armazenamento `tank`.

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

O novo sistema de arquivos do ZFS, `tank/fs`, pode utilizar tanto espaço em disco quanto for necessário e é automaticamente montando em `/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

Normalmente, é provável que você queira criar e organizar uma hierarquia de sistemas de arquivos que corresponda às suas necessidades organizacionais. Para obter mais informações sobre a criação de uma hierarquia de sistemas de arquivos do ZFS, consulte [“Criando uma hierarquia de sistemas de arquivos ZFS” na página 56](#).

Criando um pool de armazenamento do ZFS

O exemplo anterior ilustra a simplicidade do ZFS. O restante deste capítulo apresenta um exemplo mais completo, semelhante ao que poderia ser encontrado em seu ambiente. A primeira tarefa consiste em identificar os seus requisitos de armazenamento e criar um pool de armazenamento. O pool descreve as características físicas do armazenamento e deve ser criado antes dos sistemas de arquivos.

▼ Como identificar os requisitos de armazenamento para o pool de armazenamento do ZFS

1 Determine os dispositivos disponíveis para o conjunto de armazenamento.

Antes de criar um pool de armazenamento, você deve determinar que dispositivos armazenarão os dados. Esses dispositivos devem ser discos de no mínimo 128 MB e não devem estar sendo utilizados por outras partes do sistema operacional. Os dispositivos podem ser faixas individuais de um disco pré-formatado ou podem ser discos inteiros que o ZFS formata como uma única e grande faixa.

No exemplo de armazenamento utilizado em [“Como criar um pool de armazenamento do ZFS” na página 55](#), suponha que os discos `/dev/dsk/c1t0d0` e `/dev/dsk/c2t0d0` estejam totalmente disponíveis para uso.

Para obter mais informações sobre os discos e como são usados e rotulados, consulte [“Usando discos em um pool de armazenamento do ZFS” na página 65](#).

2 Escolha a replicação de dados.

O ZFS oferece suporte a vários tipos de replicação de dados, o que determina que tipos de falhas de hardware o conjunto pode suportar. O ZFS oferece suporte a configurações não redundantes (distribuídas), bem como a espelhamento e RAID-Z (uma variação de RAID-5).

No exemplo de armazenamento em [“Como criar um pool de armazenamento do ZFS” na página 55](#), utiliza-se o espelhamento básico de dois discos disponíveis.

Para obter mais informações sobre os recursos de replicação do ZFS, consulte [“Recursos de replicação de um pool de armazenamento do ZFS” na página 69](#).

▼ Como criar um pool de armazenamento do ZFS

1 Torne-se raiz e adote uma função equivalente com os perfis corretos do ZFS apropriado.

Para obter mais informações sobre os perfis corretos do ZFS, consulte [“Perfis de direitos do ZFS” na página 287](#).

2 Escolha um nome para o conjunto de armazenamento.

O nome do conjunto é utilizado para identificar o conjunto de armazenamento quando os comandos `zpool` ou `zfs` estão sendo utilizados. A maioria dos sistemas requer somente um único conjunto, tornando possível selecionar o nome de sua preferência, contanto que esteja de acordo com os requisitos de identificação citados em [“Requisitos para nomeação de componentes do ZFS” na página 50](#).

3 Crie o pool.

Por exemplo, o comando a seguir cria um conjunto espelhado chamado `tank`:

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

Se um ou mais dispositivos contiverem outro sistema de arquivos ou estiverem em uso, o comando não poderá criar o pool.

Para obter mais informações sobre a criação de pools de armazenamento, consulte [“Criando um pool de armazenamento do ZFS” na página 72](#). Para obter mais informações sobre como é determinado o uso do dispositivo, consulte [“Detectando dispositivos em uso” na página 78](#).

4 Visualize os resultados.

Você pode determinar se seu pool foi criado com sucesso usando o comando `zpool list`.

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

Para obter mais informações sobre a visualização do status do pool, consulte [“Consultando status de pool de armazenamento do ZFS” na página 106](#).

Criando uma hierarquia de sistemas de arquivos ZFS

Depois de criar um pool de armazenamento para armazenar dados, você pode criar sua hierarquia de sistemas de arquivos. As hierarquias são mecanismos simples, porém, eficazes de organização de informações. São muito familiares àqueles que já usaram sistemas de arquivos.

O ZFS permite que os sistemas de arquivos sejam organizados em hierarquias, nas quais cada sistema de arquivo tem apenas um único pai. A raiz da hierarquia é sempre o nome do pool. O ZFS favorece esta hierarquia através do suporte à herança de propriedade para que as propriedades comuns possam ser definidas rápida e facilmente em árvores inteiras dos sistemas de arquivos.

▼ Como determinar a hierarquia do sistema de arquivos ZFS

1 Selecionar a granularidade do sistema de arquivos.

Os sistemas de arquivos ZFS são o ponto central da administração. São leves e podem ser criados facilmente. Um bom modelo a ser utilizado é o de estabelecer um sistema de arquivos por usuário ou projeto, já que este modelo permite que as propriedades, instantâneos e backups possam ser controlados em uma base por usuário ou por projeto.

Os dois sistemas de arquivos ZFS, `bonwick` e `billm`, são criados em [“Como criar sistemas de arquivos ZFS” na página 57](#).

Para obter mais informações sobre o gerenciamento de sistemas de arquivos, consulte [Capítulo 6, “Gerenciando sistemas de arquivos ZFS do Oracle Solaris”](#).

2 Agrupe os sistemas de arquivos semelhantes.

O ZFS permite que os sistemas de arquivos sejam organizados em hierarquias para que os sistemas de arquivos semelhantes possam ser agrupados. Esse modelo oferece um ponto central de administração para o controle de propriedades e a administração de sistemas de arquivos. Os sistemas de arquivos semelhantes devem ser criados sob um mesmo nome.

No exemplo em [“Como criar sistemas de arquivos ZFS” na página 57](#), os dois sistemas de arquivos são inseridos em um sistema de arquivos chamado `home`.

3 Escolha as propriedades do sistema de arquivos.

A maioria das características do sistema de arquivos é controlada por propriedades. Essas propriedades controlam vários comportamentos, incluindo onde os sistemas de arquivos são montados, como são compartilhados, se utilizam compactação e se as cotas estão ativas.

No exemplo em [“Como criar sistemas de arquivos ZFS” na página 57](#), todos os diretórios `home` estão montados no usuário `/export/zfs/`, compartilhados através do NFS e apresentam a compactação ativada. Além disso, no usuário `bonwick`, é obrigatória uma cota de 10 GB.

Para obter mais informações sobre propriedades, consulte [“Introduzindo as propriedades do ZFS” na página 187](#).

▼ Como criar sistemas de arquivos ZFS

1 Torne-se raiz e adote uma função equivalente com os perfis corretos do ZFS apropriado.

Para obter mais informações sobre os perfis corretos do ZFS, consulte [“Perfis de direitos do ZFS” na página 287](#).

2 Crie a hierarquia desejada.

Neste exemplo, é criado um sistema de arquivos que age como um contêiner para sistemas de arquivos individuais.

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

3 Defina as propriedades de herança.

Depois que a hierarquia do sistema de arquivos tiver sido estabelecida, defina as propriedades que devem ser compartilhadas com os usuários:

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

NAME	PROPERTY	VALUE	SOURCE
tank/home	compression	on	local

Você pode definir as propriedades do sistema de arquivos quando o mesmo é criado. Por exemplo:

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

Para obter mais informações sobre propriedades e herança de propriedades, consulte [“Introduzindo as propriedades do ZFS” na página 187](#).

Em seguida, os sistemas de arquivos individuais são agrupados no sistema de arquivos home no pool tank.

4 Crie os sistemas de arquivos individuais.

Observe que os sistemas de arquivos podem ter sido criados e, então, as propriedades podem ter sido alteradas no nível home. Todas as propriedades podem ser alteradas dinamicamente enquanto os sistemas de arquivos estão sendo usados.

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

Esses sistemas de arquivos herdam os valores de propriedade dos seus pais, e são montados automaticamente no `/export/zfs/ usuário` e são compartilhados por NFS. Não é necessário editar o arquivo `/etc/vfstab` ou o `/etc/dfs/dfstab`.

Para obter mais informações sobre a criação de sistemas de arquivos, consulte [“Criando um sistema de arquivos ZFS” na página 184](#).

Para mais informações sobre a montagem e o compartilhamento de sistemas de arquivos, consulte [“Montando e compartilhando sistemas de arquivos ZFS” na página 208](#).

5 Defina as propriedades específicas do sistema de arquivos.

Neste exemplo, uma cota de 10 GB é atribuída ao usuário `bonwick`. Essa propriedade limita a quantidade de espaço que o usuário pode utilizar, independentemente do espaço em disco disponível no pool.

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

6 Visualize os resultados.

Visualize as informações disponíveis do sistema de arquivos usando o `zfs list`:

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

Observe que o usuário `bonwick` tem somente 10 GB de espaço disponível, enquanto que o usuário `billm` pode utilizar o conjunto completo (67 GB).

Para obter mais informações sobre a visualização do status do sistema de arquivos, consulte [“Consultando informações sobre o sistema de arquivos ZFS” na página 201](#).

Para obter mais informações sobre como o espaço é utilizado e calculado, consulte [“Contabilidade de espaço em disco do ZFS” na página 62](#).

Diferenças entre o sistema de arquivos tradicional e o ZFS do Oracle Solaris

Este capítulo aborda algumas das principais diferenças existentes entre o ZFS do Oracle Solaris e os sistemas de arquivos tradicionais. Compreender essas diferenças-chaves podem ajudar a diminuir as confusões ao trabalhar com ferramentas tradicionais na interação com o ZFS.

Este capítulo traz as seguintes seções:

- “Granularidade do sistema de arquivos ZFS” na página 61
- “Contabilidade de espaço em disco do ZFS” na página 62
- “Comportamento por espaço excedido” na página 62
- “Montando sistemas de arquivos ZFS” na página 63
- “Gerenciamento de volume tradicional” na página 63
- “Novo modelo de ACL do Solaris” na página 63

Granularidade do sistema de arquivos ZFS

Historicamente, sistemas de arquivos foram restritos a um único dispositivo e, com isso, para o tamanho desse dispositivo. A criação e recriação de sistemas de arquivos tradicionais, devido às restrições do tamanho, exigem muito tempo e são muitas vezes difíceis. Os produtos de gerenciamento de volume tradicionais ajudaram a gerenciar este processo.

Por não estarem limitados a dispositivos específicos, os sistemas de arquivos ZFS podem ser criados rápida e facilmente, semelhante à criação de diretórios. Os sistemas de arquivos do ZFS crescem automaticamente dentro do espaço em disco alocado para o conjunto de armazenamento no qual residem.

Em vez de criar um sistema de arquivos, como o `/export/home`, para gerenciar vários subdiretórios de usuários, você pode criar um sistema de arquivos por usuário. É possível configurar e gerenciar facilmente muitos sistemas de arquivos aplicando propriedades que podem ser herdadas pelos sistemas de arquivos descendentes contidos na hierarquia.

Para ver um exemplo de como criar uma hierarquia de sistema de arquivos, consulte “[Criando uma hierarquia de sistemas de arquivos ZFS](#)” na página 56.

Contabilidade de espaço em disco do ZFS

O ZFS tem base no conceito de armazenamento por conjunto. Ao contrário dos típicos sistemas de arquivos, que são mapeados para armazenamentos físicos, todos os sistemas de arquivos ZFS de um pool compartilham o armazenamento disponível no pool. Desse modo, o espaço disponível relatado por utilitários como o `df` pode alterar mesmo quando o sistema de arquivos está inativo, já que outros sistemas de arquivos do conjunto consomem e liberam espaço.

Observe que o tamanho máximo do sistema de arquivos pode ser limitado pelo uso de cotas. Para obter mais informações sobre as cotas, consulte “[Definindo cotas em sistemas de arquivos ZFS](#)” na página 216. Uma quantidade especificada de espaço em disco pode ser garantida para um sistema de arquivos utilizando as reservas. Para obter mais informações sobre as reservas, consulte “[Definindo reservas nos sistemas de arquivos ZFS](#)” na página 219. Esse modelo é semelhante ao modelo de NFS, no qual vários diretórios são montados a partir do mesmo sistema de arquivos (considerar `/home`).

Todos os metadados no ZFS estão alocados dinamicamente. A maioria dos outros sistemas de arquivos pré-alocam muitos de seus metadados. Como resultado, na hora da criação do sistema de arquivos, há um custo imediato de espaço para esses metadados. Esse comportamento denota também que o número total de arquivos suportado pelos sistemas de arquivos é predeterminado. Por alocar seus metadados conforme precisa deles, o ZFS não requer quantidade de espaço inicial, e o número de arquivos está limitado somente pelo espaço em disco disponível. A saída do comando `df -g` não deve ser interpretada da mesma forma para o ZFS e para outros sistemas de arquivos. Os arquivos totais relatados são somente uma estimativa baseada na quantidade de armazenamento disponível no pool.

O ZFS é um sistema de arquivos transacional. A maioria das modificações do sistema de arquivos é incorporada dentro de grupos transacionais e é enviada ao disco assincronicamente. Antes de estarem comprometidas com o disco, essas modificações são denominadas *alterações pendentes*. A quantidade de espaço utilizado, disponível e referenciada pelo arquivo ou sistema de arquivos não considera as alterações pendentes. As alterações pendentes são consideradas em geral depois de alguns segundos. Mesmo comprometendo uma alteração no disco utilizando `fsync (3c)` ou `O_SYNC`, isso não garante necessariamente que as informações sobre o uso do espaço em disco sejam atualizadas imediatamente.

Para detalhes adicionais sobre o consumo de espaço em disco do ZFS reportado pelos comandos `du` e `df`, consulte:

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

Comportamento por espaço excedido

Os instantâneos de sistemas de arquivos são baratos e fáceis de criar no ZFS. Os instantâneos são comuns na maioria dos ambientes do ZFS. Para obter informações sobre os instantâneos do ZFS, consulte o [Capítulo 7](#), “[Trabalhando com instantâneos e clones do ZFS do Oracle Solaris](#)”.

A presença de instantâneos pode provocar alguns comportamentos inesperados ao tentar liberar espaço em disco. Normalmente, com as permissões apropriadas, é possível remover um arquivo de todo um sistema de arquivos, e esta ação resulta em mais espaço em disco disponível no sistema de arquivos. No entanto, se o arquivo a ser removido existir no instantâneo do sistema de arquivos, então nenhum espaço em disco é liberado com a exclusão do arquivo. Os blocos usados pelo arquivo continuam a ser referenciados a partir do instantâneo.

Como resultado, a exclusão do arquivo pode consumir mais espaço em disco, pois uma nova versão do diretório precisa ser criada para refletir o novo estado do espaço de nome. Este comportamento significa que é possível receber um erro ENOSPC ou EDQUOT inesperado ao tentar remover o arquivo.

Montando sistemas de arquivos ZFS

O ZFS reduz a complexidade e facilita a administração. Por exemplo, com sistemas de arquivos tradicionais, você deve editar o arquivo `/etc/vfstab` sempre que um novo sistema de arquivos for adicionado. O ZFS eliminou essa necessidade montando e desmontando automaticamente os sistemas de arquivos de acordo com as propriedades do conjunto de dados. Você não precisa gerenciar as entradas do ZFS no arquivo `/etc/vfstab`.

Para obter mais informações sobre a montagem e o compartilhamento de sistemas de arquivos ZFS, consulte [“Montando e compartilhando sistemas de arquivos ZFS” na página 208](#).

Gerenciamento de volume tradicional

De acordo com o descrito em [“Armazenamento de ZFS em pool” na página 45](#), o ZFS dispensa a necessidade de usar um gerenciador de volumes diferente. O ZFS opera em dispositivos básicos, de modo que é possível criar um pool de armazenamento constituído de volumes lógicos, tanto no software quanto no hardware. Essa configuração não é recomendada, já que o ZFS funciona melhor quando usa dispositivos físicos básicos. O uso de volumes lógicos pode prejudicar o desempenho, a segurança, ou ambos, e deve ser evitado.

Novo modelo de ACL do Solaris

As versões anteriores do sistema operacional Solaris ofereciam suporte a uma implementação de ACL baseada principalmente na especificação da ACL do esquema POSIX. As ACLs baseadas no esquema POSIX são usadas para proteger os arquivos UFS. O novo modelo de ACL do Solaris com base na especificação NFSv4 é utilizado para proteger os arquivos do ZFS.

As principais diferenças do novo modelo da ACL do Solaris são as seguintes:

- O modelo tem base na especificação do NFSv4 e é similar às ACLs de estilo NT.

- Este modelo fornece um conjunto de privilégios de acesso mais granular.
- ACLs são definidas e exibidas com os comandos `chmod` e `ls` em vez dos comandos `setfacl` e `getfacl`.
- Semânticas mais ricas designam como os privilégios de acesso são aplicados do diretório aos subdiretórios, e assim por diante.

Para obter mais informações sobre o uso de ACLs com arquivos do ZFS, consulte o [Capítulo 8](#), “Utilizando ACLs para proteger arquivos ZFS do Oracle Solaris”.

Gerenciando conjuntos de armazenamento ZFS do Oracle Solaris

Este capítulo descreve como criar e administrar conjuntos de armazenamento no ZFS do Oracle Solaris.

Este capítulo traz as seguintes seções:

- “Componentes de um pool de armazenamento do ZFS” na página 65
- “Recursos de replicação de um pool de armazenamento do ZFS” na página 69
- “Criando e destruindo pools de armazenamento do ZFS” na página 72
- “Gerenciando dispositivos em pools de armazenamento do ZFS” na página 82
- “Gerenciando propriedades do pool de armazenamento do ZFS” na página 103
- “Consultando status de pool de armazenamento do ZFS” na página 106
- “Migrando pools de armazenamento do ZFS” na página 115
- “Atualizando pools de armazenamento do ZFS” na página 121

Componentes de um pool de armazenamento do ZFS

As seções seguintes fornecem informações detalhadas sobre os componentes de pool de armazenamento:

- “Usando discos em um pool de armazenamento do ZFS” na página 65
- “Usando faixas em um pool de armazenamento do ZFS” na página 67
- “Usando arquivos em um pool de armazenamento do ZFS” na página 68

Usando discos em um pool de armazenamento do ZFS

O elemento mais básico de um conjunto de armazenamento é um armazenamento físico. O armazenamento físico pode ser qualquer dispositivo de bloco com um tamanho de pelo menos 128 MB. Geralmente, este dispositivo é um disco rígido que é visível para o sistema no diretório `/dev/dsk`.

Um dispositivo de armazenamento que pode ser um disco inteiro (`c1t0d0`) ou uma faixa individual (`c0t0d0s7`). O modo recomendado de operação é utilizar um disco inteiro, caso em que o disco não necessita ser especialmente formatado. O ZFS formata o disco usando uma legenda EFI para conter uma única faixa grande. Quando usado desta forma, a tabela de partição que é exibida pelo comando `format` aparece semelhante à seguinte:

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

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

Para utilizar um disco inteiro, o disco deve ser nomeado utilizando a convenção de identificação `/dev/dsk/cXtXdXsX`. Alguns drivers de terceiros usam uma convenção de nomeação diferente ou colocam o disco em um local diferente do diretório `/dev/dsk`. Para usar estes discos, é necessário rotular manualmente o disco e fornecer uma faixa para o ZFS.

O ZFS aplica uma legenda EFI quando você cria um pool de armazenamento com discos inteiros. Para obter mais informações sobre legendas EFI, consulte [“EFI Disk Label” no *System Administration Guide: Devices and File Systems*](#).

Um disco definido para um pool raiz ZFS, precisa ser criado com uma legenda SMI e não uma legenda EFI. Você pode relegendar um disco com uma legenda SMI usando o comando `format -e`.

Os discos podem ser especificados com o uso do caminho completo, como `/dev/dsk/c1t0d0`, ou de um nome abreviado que consiste no nome do dispositivo dentro do diretório `/dev/dsk`, como `c1t0d0`. Abaixo encontram-se, por exemplo, nomes de disco válidos:

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

A utilização de discos inteiros é a forma mais simples de criar conjuntos de armazenamento ZFS. As configurações do ZFS se tornam progressivamente mais complexas, dos pontos de vista de gerenciamento, confiabilidade e desempenho, quando você constrói pools a partir de faixas de disco, LUNs em matrizes de hardware RAID ou em volumes apresentados pelos gerenciadores de volume baseados em software. As considerações abaixo podem ajudar você a determinar como configurar o ZFS com outras soluções de armazenamento de hardware ou software:

- Se construir as configurações do ZFS em cima de LUNs a partir de matrizes de hardware RAID, é necessário entender a relação entre os recursos de redundância ZFS e os recursos de redundância oferecidos pela matriz. Determinadas configurações podem oferecer redundância e desempenho adequados, mas outras configurações não.
- Você pode construir dispositivos lógicos para o ZFS usando volumes apresentados pelos gerenciadores de volume baseados em software, como Solaris Volume Manager (SVM) ou Veritas Volume Manager (VxVM). No entanto, estas configurações não são recomendáveis. Embora o ZFS funcione corretamente nestes dispositivos, o resultado pode ser um desempenho inferior a ótimo.

Para obter informações adicionais sobre as recomendações de pool de armazenamento, consulte o site de boas práticas do ZFS:

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

Os discos são identificados pelo caminho e pelo ID do dispositivo, se disponível. Em sistemas onde a informação de ID do dispositivo está disponível, este método de identificação permite que dispositivos sejam configurados sem atualizar o ZFS. Como a geração e gerenciamento de dispositivo de ID pode variar de acordo com o sistema, exporte o conjunto primeiro antes de mover dispositivos, assim como mover um disco de um controlador para outro. Um evento do sistema, como uma atualização de firmware ou outra alteração de hardware, pode alterar as IDs do dispositivo no conjunto de armazenamento ZFS, que pode causar a invalidação do dispositivo.

Usando faixas em um pool de armazenamento do ZFS

Os discos podem ser rotulados com uma legenda Solaris VTOC (SMI) tradicional quando você cria um pool de armazenamento com uma faixa de disco.

Para um conjunto raiz ZFS inicializável, os discos no conjunto devem conter segmentos e precisam ser legendados com uma legenda SMI. A configuração mais simples é colocar toda a capacidade do disco na faixa 0 e usar essa faixa para o pool raiz.

Em um sistema com base em SPARC, um disco de 72 GB possui 68 GB de espaço útil localizados no segmento 0, conforme mostrado na seguinte saída de format:

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

Part      Tag      Flag      Cylinders      Size      Blocks
```

0	root	wm	0 - 14086	68.35GB	(14087/0/0)	143349312
1	unassigned	wm	0	0	(0/0/0)	0
2	backup	wm	0 - 14086	68.35GB	(14087/0/0)	143349312
3	unassigned	wm	0	0	(0/0/0)	0
4	unassigned	wm	0	0	(0/0/0)	0
5	unassigned	wm	0	0	(0/0/0)	0
6	unassigned	wm	0	0	(0/0/0)	0
7	unassigned	wm	0	0	(0/0/0)	0

Em um sistema com base em x86, um disco de 72 GB possui 68 GB de espaço útil localizados no segmento 0, conforme mostrado na seguinte saída de `format`. A faixa 8 contém uma pequena quantidade de informações de inicialização. A faixa 8 não precisa de administração e não pode ser alterada.

```
# format
```

```
.
```

```
selecting clt0d0
```

```
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

Usando arquivos em um pool de armazenamento do ZFS

O ZFS também permite que usar arquivos do UFS como dispositivos virtuais no pool de armazenamento. Este recurso se destina principalmente a testar e capacitar experimentos simples, não para uso em produção. O motivo é que **qualquer uso de arquivos depende do sistema de arquivos subjacente para ter consistência**. Se criar um pool do ZFS baseado em arquivos de um sistema de arquivos UFS, você estará implicitamente confiando no UFS para garantir correção e semântica síncrona.

No entanto, os arquivos podem ser bastante úteis ao utilizar o ZFS pela primeira vez ou experimentar configurações mais complicadas quando dispositivos físicos insuficientes estão presentes. É necessário especificar todos os arquivos como caminhos completos e estes devem ter tamanho de pelo menos 64 MB.

Recursos de replicação de um pool de armazenamento do ZFS

O ZFS fornece redundância de dados, bem como propriedades de autocorreção, em uma configuração espelhada e RAID-Z.

- “Configuração de pool de armazenamento espelhado” na página 69
- “Configuração de pool de armazenamento RAID-Z” na página 69
- “Dados de autocorreção em uma configuração redundante” na página 71
- “Distribuição dinâmica em um pool de armazenamento” na página 71
- “Pool armazenamento híbrido do ZFS” na página 71

Configuração de pool de armazenamento espelhado

Uma configuração de pool de armazenamento espelhado requer pelo menos dois discos, de preferência em controladores separados. Vários discos podem ser usados em uma configuração espelhada. Além disso, você pode criar mais de um espelho em cada pool. Conceitualmente, uma configuração espelhada simples teria a aparência seguinte:

```
mirror c1t0d0 c2t0d0
```

Conceitualmente, uma configuração espelhada mais complexa teria a aparência seguinte:

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

Para obter mais informações sobre a criação de pool de armazenamento espelhado, consulte “Criando um pool de armazenamento espelhado” na página 72.

Configuração de pool de armazenamento RAID-Z

Além de uma configuração de conjunto em armazenamento espelhado, o ZFS fornece uma configuração RAID-Z com tolerância de falha de paridade única, dupla ou tripla. RAID-Z de paridade dupla (`raidz` ou `raidz1`) é semelhante ao RAID-5. O RAID-Z de paridade dupla (`raidz2`) é semelhante ao RAID-6.

Para mais informações sobre RAIDZ-3 (`raidz3`), consulte o blog a seguir:

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

Todos os algoritmos tradicionais semelhantes ao RAID-5 (RAID-4, RAID-6, RDP e EVEN-ODD, por exemplo) podem apresentar um problema conhecido como "buraco de gravação de RAID-5". Se apenas parte de uma faixa RAID-5 for gravado, e a potência se perder antes de todos os blocos terem sido registrados no disco, a paridade permanecerá dessincronizada com os dados e, portanto, ficará inutilizada para sempre (a não ser que uma gravação subsequente de faixa inteira a substitua). No RAID-Z, o ZFS usa faixas RAID de

largura variável, de forma que todas as gravações são gravações de faixa inteira. Este design é possível apenas porque o ZFS integra o gerenciamento do sistema de arquivos e do dispositivo de tal forma que os metadados do sistema de arquivos têm informações suficientes sobre o modelo de redundância de dados subjacente para lidar com faixas RAID de largura variável. O RAID-Z é a primeira solução somente de software do mundo para o "buraco de gravação de RAID-5".

Uma configuração RAID-Z com N discos de tamanho X com discos de paridade P podem conter aproximadamente $(N-P)*X$ bytes e podem resistir a P dispositivo(s) falho(s) antes que a integridade dos dados seja comprometida. São necessários pelo menos dois discos para uma configuração RAID-Z de paridade única e pelo menos três discos para uma configuração RAID-Z de paridade dupla. Por exemplo, se houver três discos em uma configuração RAID-Z de paridade única, os dados da paridade ocuparão espaço igual a um dos três discos. Do contrário, nenhum hardware especial será necessário para criar uma configuração RAID-Z.

Conceitualmente, uma configuração RAID-Z com três discos teria a aparência semelhante à seguinte:

```
raidz c1t0d0 c2t0d0 c3t0d0
```

Conceitualmente, uma configuração espelhada RAID-Z mais complexa teria a aparência a seguir:

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

Se estiver criando uma configuração RAID-Z com muitos discos, considere dividir os discos em vários agrupamentos. Por exemplo, uma configuração RAID-Z com 14 discos é melhor dividido em 7 agrupamentos de disco. As configurações RAID-Z com agrupamentos de dígito único de discos têm um melhor desempenho.

Para informações sobre a criação de um conjunto de armazenamento RAID-Z, consulte [“Criando um conjunto de armazenamento RAID-Z”](#) na página 74.

Para informações sobre a escolha entre uma configuração espelhada ou uma configuração RAID-Z levando em consideração desempenho e espaço, consulte a entrada de blog a seguir:

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

Para informações adicionais sobre as recomendações de conjunto de armazenamento RAID-Z, consulte o site de boas práticas do ZFS:

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

Pool armazenamento híbrido do ZFS

O conjunto de armazenamento híbrido do ZFS, disponível na série de produtos Sun Storage 7000, é um conjunto de armazenamento especial que combina DRAM, SSDs e HDDs para melhorar o desempenho e aumentar a capacidade, ao mesmo tempo que reduzir o consumo de energia. Com esta interface de gerenciamento de produto, é possível selecionar a configuração de redundância do ZFS do conjunto de armazenamento e gerenciar facilmente outras opções de configuração.

Para obter mais informações sobre este produto, consulte o *Sun Storage Unified Storage System Administration Guide*.

Dados de autocorreção em uma configuração redundante

O ZFS fornece dados de autocorreção em uma configuração espelhada ou RAID-Z.

Quando um bloco de dados incorretos é detectado, o ZFS não apenas tira os dados corretos de uma cópia redundante como também repara os dados incorretos substituindo-os pela cópia correta.

Distribuição dinâmica em um pool de armazenamento

O ZFS fatia os dados dinamicamente através de todos os dispositivos virtuais de nível superior. A decisão sobre o local o qual colocar os dados é tomada na hora da gravação, de forma que nenhuma faixa de largura fixa é criada no momento da alocação.

Quando novos dispositivos virtuais são adicionados a um conjunto, o ZFS aloca dados gradualmente no novo dispositivo com o objetivo de manter o desempenho e políticas de alocação de espaço. Cada dispositivo virtual também pode ser um dispositivo de espelho ou RAID-Z que contém outros dispositivos de disco ou arquivos. Esta configuração oferece flexibilidade no controle das características de falha no conjunto. É possível criar, por exemplo, as configurações a seguir com quatro discos:

- Quatro discos que usem distribuição dinâmica
- Uma configuração RAID-Z de quatro direções
- Dois espelhos bidirecionais que usem distribuição dinâmica

Apesar do ZFS suportar a combinação de diferentes tipos de dispositivos virtuais dentro do mesmo conjunto, evite esta prática. Você pode criar, por exemplo, um pool com um espelho bidirecional e uma configuração RAID-Z tridirecional. No entanto, a tolerância a falhas é tão boa quanto a do seu pior dispositivo virtual, RAID-Z neste caso. A melhor prática é utilizar dispositivos virtuais de nível superior do mesmo tipo com o mesmo nível de redundância em cada dispositivo.

Criando e destruindo pools de armazenamento do ZFS

As seções seguintes descrevem diferentes situações de criação e destruição de pools de armazenamento do ZFS:

- “Criando um pool de armazenamento do ZFS” na página 72
- “Exibindo informações do dispositivo virtual do pool de armazenamento” na página 77
- “Manipulando erros de criação do pool de armazenamento do ZFS” na página 78
- “Destruindo pools de armazenamento do ZFS” na página 81

Criar e destruir conjuntos é rápido e fácil. Entretanto, seja cuidadoso ao efetuar estas operações. Embora sejam realizadas verificações para impedir o uso de dispositivos que estão sendo usados em um novo pool, o ZFS nem sempre sabe quando um dispositivo está sendo utilizado. Destruir um conjunto é mais fácil do que construir um. Use `zpool destroy` com cuidado. Este comando simples tem consequências significantes.

Criando um pool de armazenamento do ZFS

Para criar um pool de armazenamento, use o comando `zpool create`. Este comando adota um nome de pool e qualquer quantidade de dispositivos virtuais como argumentos. O nome do conjunto deve cumprir as convenções de identificação descritas em “Requisitos para nomeação de componentes do ZFS” na página 50.

Criando um pool de armazenamento básico

O seguinte comando cria um novo pool denominado `tank` que contém os discos `c1t0d0` e `c1t1d0`:

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

Nomes de dispositivos representando discos inteiros são encontrados no diretório `/dev/dsk` e estão legendados adequadamente pelo ZFS para conter um único e grande segmento. Os dados são distribuídos dinamicamente nos dois discos.

Criando um pool de armazenamento espelhado

Para criar um pool espelhado, use a palavra-chave `mirror`, seguida de uma quantidade de dispositivos de armazenamento que formarão o espelho. É possível especificar vários espelhos com a repetição da palavra-chave `mirror` na linha de comando. O comando abaixo cria um pool com dois espelhos bidirecionais:

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

A segunda palavra-chave `mirror` indica que um novo dispositivo virtual de primeiro nível está sendo especificado. Os dados são distribuídos dinamicamente nos dois espelhos e redundados adequadamente em cada disco.

Para obter mais informações sobre as configurações espelhadas recomendadas, vá ao site:

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

No momento, as operações a seguir são suportadas em uma configuração espelhada do ZFS:

- Adicionar outro conjunto de discos de um dispositivo de nível superior adicional (vdev) a uma configuração espelhada existente. Para obter mais informações, consulte “[Adicionando dispositivos a um pool de armazenamento](#)” na página 82.
- Anexar discos adicionais a uma configuração espelhada existente. Ou, anexar discos adicionais a uma configuração não replicada para criar uma configuração espelhada. Para obter mais informações, consulte “[Anexando e desanexando dispositivos em um pool de armazenamento](#)” na página 87.
- Substituir um disco ou discos em uma configuração espelhada existente desde que os discos de substituição sejam maiores ou iguais ao tamanho do dispositivo a ser substituído. Para obter mais informações, consulte “[Substituindo dispositivos em um pool de armazenamento](#)” na página 95.
- Desanexando um disco em uma configuração espelhada desde que os dispositivos restantes forneçam redundância adequada para a configuração. Para obter mais informações, consulte “[Anexando e desanexando dispositivos em um pool de armazenamento](#)” na página 87.
- Dividindo uma configuração espelhada ao desanexar um dos discos ao criar um novo conjunto idêntico. Para mais informações, consulte “[Criando um novo conjunto ao dividir um conjunto de armazenamento ZFS espelhado](#)” na página 89.

Não é possível remover completamente um dispositivo que não é um dispositivo de log ou cache de um conjunto de armazenamento espelhado. Uma RFE é preenchida para esse recurso.

Criando um pool raiz ZFS

É possível instalar e inicializar a partir de um sistema de arquivos ZFS. Consulte as seguintes informações de configuração do pool raiz:

- Os discos usados para o pool raiz devem ter uma legenda VTOC (SMI) e o pool deve ser criado com faixas de disco.
- O conjunto raiz deve ser criado como uma configuração espelhada ou uma configuração de disco único. Não é possível adicionar discos adicionais pra criar vários dispositivos virtuais espelhados de nível superior ao utilizar o comando `zpool add`, mas é possível expandir um dispositivo virtual espelhado ao utilizar o comando `zpool attach`.
- A configuração distribuída ou RAID-Z não é suportada.
- Um conjunto raiz não pode possuir um dispositivo de log em separado.
- Se tentar utilizar uma configuração não suportada para um conjunto raiz, serão exibidas mensagens semelhantes às seguintes:

```
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
```

Para mais informações sobre a instalação e a inicialização de um sistema de arquivos raiz ZFS, consulte [Capítulo 5, “Instalando e inicializando um sistema de arquivos raiz ZFS do Oracle Solaris”](#).

Criando um conjunto de armazenamento RAID-Z

A criação de um pool RAID-Z de paridade única é idêntica à criação de um pool espelhado, exceto que a palavra-chave `raidz` ou `raidz1` é usada em lugar de `mirror`. O exemplo abaixo mostra como criar um pool com um dispositivo RAID-Z único que está composto por dois discos:

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

Este exemplo ilustra que os discos podem ser especificados ao utilizar seus nomes de dispositivo de estenografia ou seus nomes de dispositivos completos. Tanto `/dev/dsk/c5t0d0` quanto `c5t0d0` fazem referência ao mesmo disco.

É possível criar configurações RAID-Z de paridade dupla ou tripla ao utilizar a palavra-chave `raidz2` ou `raidz3` ao criar o conjunto. Por exemplo:

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

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

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

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

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
raidz3-0	ONLINE	0	0	0
c0t0d0	ONLINE	0	0	0
c1t0d0	ONLINE	0	0	0

```

c2t0d0 ONLINE      0      0      0
c3t0d0 ONLINE      0      0      0
c4t0d0 ONLINE      0      0      0
c5t0d0 ONLINE      0      0      0
c6t0d0 ONLINE      0      0      0
c7t0d0 ONLINE      0      0      0

```

errors: No known data errors

No momento, as operações a seguir são suportadas em uma configuração RAID-Z do ZFS:

- Adicionar outro conjunto de discos para um dispositivo virtual de nível superior adicional em uma configuração RAID-Z existente. Para obter mais informações, consulte [“Adicionando dispositivos a um pool de armazenamento” na página 82.](#)
- Substituir um disco ou discos em uma configuração RAID-Z existente desde que os discos de substituição sejam maiores ou iguais ao dispositivo a ser substituído. Para obter mais informações, consulte [“Substituindo dispositivos em um pool de armazenamento” na página 95.](#)

No momento, as operações seguintes *não* são suportadas em uma configuração RAID-Z:

- Anexação de um disco adicional a uma configuração RAID-Z existente.
- Desanexação de um disco de uma configuração RAID-Z, exceto quando estiver desanexando um disco que foi substituído por um disco sobressalente.
- Não é possível remover completamente um dispositivo que não é um dispositivo de log ou de cache de uma configuração RAID-Z. Uma RFE é preenchida para esse recurso.

Para obter informações sobre a configuração RAID-Z, consulte [“Configuração de pool de armazenamento RAID-Z” na página 69.](#)

Criando um conjunto de armazenamento do ZFS com dispositivos de log

Por padrão, o ZIL é alocado dos blocos no pool principal. No entanto, pode ser possível obter melhor desempenho usando dispositivos separados de registro de intenção, como NVRAM ou um disco dedicado. Para obter mais informações sobre os dispositivos de log do ZFS, consulte [“Configurando dispositivos de log do ZFS separados” na página 33.](#)

É possível configurar um dispositivo de log ZFS quando o conjunto de armazenamento for criado ou depois da sua criação.

O exemplo a seguir mostra como criar um conjunto de armazenamento espelhado com dispositivos de log espelhados:

```

# zpool create datap mirror c1t1d0 c1t2d0 mirror c1t3d0 c1t4d0 log mirror c1t5d0 c1t8d0
# zpool status datap
  pool: datap
  state: ONLINE

```

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

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

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

Para obter informações sobre como recuperar de um dispositivo de registro com falha, consulte [Exemplo 11-2](#).

Criando um conjunto de armazenamento do ZFS com dispositivos de cache

Você pode criar um pool de armazenamento com dispositivos de cache para dados de pool de armazenamento cache. Por exemplo:

```
# 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
```

Considere os pontos a seguir ao determinar a criação de um conjunto de armazenamento ZFS com dispositivos de cache:

- A utilização dos dispositivos de cache fornecem o maior aperfeiçoamento de desempenho para cargas de trabalho aleatórias de leitura de conteúdo geralmente estático.
- A capacidade e as leituras podem ser monitoradas usando o comando `zpool iostat`.

- Um único ou vários dispositivos de cache podem ser adicionados quando o conjunto é criado. Também podem ser adicionados e removidos depois da criação do conjunto. Para obter mais informações, consulte o [Exemplo 4-4](#).
- Os dispositivos de cache não podem ser espalhados ou fazer parte de uma configuração RAID-Z.
- Se for encontrado em erro de leitura em um dispositivo de cache, aquele E/S é reemitido para o dispositivo de conjunto de armazenamento original, que pode fazer parte de uma configuração espelhada ou RAID-Z. O conteúdo dos dispositivos de cache é considerado volátil, similar a outros caches de sistema.

Exibindo informações do dispositivo virtual do pool de armazenamento

Cada conjunto de armazenamento contém um ou mais dispositivos virtuais. Um *dispositivo virtual* é uma representação interna do conjunto de armazenamento que descreve o layout do armazenamento físico e suas características de falha. Como tal, um dispositivo virtual representa os dispositivos de disco ou arquivos que são usados para criar o pool de armazenamento. Um conjunto pode ter qualquer número de dispositivos virtuais na parte superior da configuração, conhecida como *top-level vdev*.

Se o dispositivo virtual de nível superior contém dois ou mais dispositivos físicos, a configuração fornece redundância de dados como dispositivos de espelho ou virtual RAID-Z. Estes dispositivos virtuais consistem em discos, faixas de discos ou arquivos. Um sobressalente é um dispositivo virtual especial que rastreia dispositivos de reserva disponíveis para um conjunto.

Os exemplos a seguir mostram como criar um conjunto que consiste em dois dispositivos virtuais de nível superior, cada um com espelho de dois discos:

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

O exemplo abaixo mostra como criar um conjunto que consiste em um dispositivo de virtual nível superior de quatro discos:

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

É possível adicionar outro dispositivo virtual de nível superior a este conjunto utilizando o comando `zpool add`. Por exemplo:

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

Os discos, segmentos de discos ou arquivos que são utilizados em conjuntos não redundantes que funcionam como dispositivos virtuais de nível superior. Os pools de armazenamento

normalmente contêm múltiplos dispositivos virtuais de primeiro nível. O ZFS cria dinamicamente faixas de dados entre todos os dispositivos virtuais de primeiro nível em um pool.

Os dispositivos virtuais e os dispositivos físicos contidos no pool de armazenamento do ZFS são exibidos com o comando `zpool status`. Por exemplo:

```
# 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
```

Manipulando erros de criação do pool de armazenamento do ZFS

Os erros de criação de pool podem ocorrer por vários motivos. Alguns desses motivos são óbvios, como quando um dispositivo especificado não existe, enquanto outros são mais sutis.

Detectando dispositivos em uso

Antes de formatar um dispositivo, o ZFS determina se o disco está sendo usado pelo ZFS ou por alguma outra parte do sistema operacional. Se o disco estiver em uso, poderão ser exibidos erros como os seguintes:

```
# 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).
```

Alguns destes erros podem ser ignorados utilizando a opção `-f`, mas grande parte deles não. As condições a seguir não podem ser ignoradas utilizando a opção `-f`, sendo necessário corrigi-los manualmente:

Sistema de arquivos montado	O disco ou uma de suas faixas contém um sistema de arquivos que está montado atualmente. Para corrigir este erro, use o comando <code>umount</code> .
Sistema de arquivos em <code>/etc/vfstab</code>	O disco contém um sistema de arquivos que está listado no arquivo <code>/etc/vfstab</code> , mas o sistema de arquivos não está montado atualmente. Para corrigir este erro, remova ou anote a linha no arquivo <code>/etc/vfstab</code> .
Dispositivo de dump dedicado	O disco está em uso como o dispositivo de dump dedicado para o sistema. Para corrigir este erro, use o comando <code>dumpadm</code> .
Parte de um pool do ZFS	O disco ou o arquivo faz parte de um pool de armazenamento ativo do ZFS. Para corrigir este erro, use o comando <code>zpool destroy</code> para destruir o pool, se este não for mais necessário. Ou, use o comando <code>zpool detach</code> para desanexar o disco do outro pool. Somente é possível desanexar um disco do pool de armazenamento espelhado.
As verificações de dispositivo em uso abaixo servem como avisos úteis e podem ser ignoradas com o uso da opção <code>-f</code> para criar o pool:	
Contém um sistema de arquivos	O disco contém um sistema de arquivos conhecido, embora não esteja montado e não pareça estar em uso.
Parte de volume	Este disco é parte de um volume do Solaris Volume Manager.
Live upgrade	O disco está em utilização como um ambiente de inicialização alternativo para o Oracle Solaris Live Upgrade.
Parte de pool exportado do ZFS	O disco faz parte de um pool de armazenamento que foi exportado ou removido manualmente de um sistema. Neste último caso, o pool é relatado como <code>potentially active</code> , uma vez que o disco pode ser ou não uma unidade anexada a uma rede em uso por outro sistema. Tome cuidado ao substituir um pool potencialmente ativo.

O exemplo abaixo mostra como a opção `-f` é usada:

```
# 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
```

O ideal é corrigir os erros, em vez de utilizar a opção `-f` para ignorá-los.

Níveis de replicação inconsistentes

Não é recomendável criar pools com dispositivos virtuais de diferentes níveis de replicação. O comando `zpool` tenta impedir que você crie por acidente um pool com níveis de redundância inconsistentes. Se tentar criar um pool com uma configuração assim, serão exibidos erros como os seguintes:

```
# 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
```

É possível ignorar estes erros com a opção `-f`, mas deve-se evitar esta prática. O comando também avisa sobre a criação de um pool espelhado ou RAID-Z que use dispositivos de diferentes tamanhos. Apesar desta configuração ser permitida, níveis de redundância incompatíveis resultam em espaço de disco inutilizado no dispositivo maior. A opção `-f` é necessária para ignorar os avisos.

Simulando uma criação de pool de armazenamento

A tentativa de criar um conjunto pode falhar de forma inesperada de diferentes maneiras, e formatar discos é uma ação potencialmente prejudicial. Para estas razões, o comando `zpool create` possui a opção adicional, `-n`, que simula a criação do conjunto sem realmente escrever no dispositivo. Esta opção *simulação* efetua a verificação do dispositivo em uso e a validação do nível de replicação, e relata quaisquer erros durante o processo. Se nenhum erro for encontrado, será exibida uma saída semelhante à seguinte:

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

    tank
      mirror
        c1t0d0
        c1t1d0
```

Alguns erros não podem ser detectados sem a criação real do pool. O exemplo mais comum é a especificação do mesmo dispositivo duas vezes na mesma configuração. Não é possível detectar este erro com certeza sem realmente gravar os dados, de forma que o comando `zpool create -n` pode relatar êxito e ainda assim falhar em criar o conjunto quando realmente executado sem esta opção.

Ponto de montagem padrão para pools de armazenamento

Quando um conjunto é criado, o ponto de montagem padrão para o conjunto de dados raiz de nível superior é `/pool-name`. Este diretório não deve existir ou deve estar vazio. Se o diretório não existir, será criado automaticamente. Se o diretório estiver vazio, o conjunto de dados raiz será montado em cima do diretório existente. Para criar um pool com um ponto de montagem padrão diferente, use a opção `-m` do comando `zpool create`. Por exemplo:

```
# 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
```

Este comando cria um novo conjunto `home` e o conjunto de dados `home` com um ponto de montagem `/export/zfs`.

Para obter mais informações sobre pontos de montagem, consulte [“Gerenciando pontos de montagem do ZFS”](#) na página 208.

Destruindo pools de armazenamento do ZFS

Os pools são destruídos com o uso do comando `zpool destroy`. Este comando destrói o pool mesmo se contiver conjuntos de dados montados.

```
# zpool destroy tank
```



Cuidado – Tome muito cuidado ao destruir um pool. Certifique-se de que está destruindo o conjunto certo e sempre tenha cópias dos dados. Se destruir por acidente o pool errado, poderá tentar recuperá-lo. Para obter mais informações, consulte [“Recuperando pools de armazenamento destruídos do ZFS”](#) na página 120.

Destruindo um pool com dispositivos falhos

O ato de destruir um conjunto requer que os dados sejam gravados em disco para indicar que o conjunto deixou de ser válido. Estas informações de estado evitam que o dispositivo apareça como um possível pool ao realizar uma importação. Se um ou mais dispositivos não estiverem disponíveis, o pool pode ainda assim ser destruído. No entanto, as informações de estado necessárias não serão gravadas nestes dispositivos indisponíveis.

Estes dispositivos, quando reparados de maneira adequada, são reportados como *potencialmente ativos* ao criar um novo conjunto. Eles aparecem como dispositivos válidos ao buscar por conjuntos para importar. Se um pool apresentar bastantes dispositivos falhos, de forma que o próprio pool é falho (o que significa que o dispositivo virtual de primeiro nível é

falho), o comando imprimirá um aviso e não poderá concluir sem a opção `-f`. Esta opção é necessária porque o conjunto não pode ser aberto, então não é possível saber se os dados estão armazenados lá. Por exemplo:

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

Para obter mais informações sobre a integridade de pools e dispositivos, consulte [“Determinando o status de integridade de pools de armazenamento do ZFS”](#) na página 112.

Para obter mais informações sobre a importação de pools, consulte [“Importando pools de armazenamento do ZFS”](#) na página 119.

Gerenciando dispositivos em pools de armazenamento do ZFS

Grande parte das informações básicas relacionadas a dispositivos é tratada em [“Componentes de um pool de armazenamento do ZFS”](#) na página 65. Depois da criação de um conjunto, é possível efetuar várias tarefas para gerenciar os dispositivos físicos dentro do conjunto.

- [“Adicionando dispositivos a um pool de armazenamento”](#) na página 82
- [“Anexando e desanexando dispositivos em um pool de armazenamento”](#) na página 87
- [“Criando um novo conjunto ao dividir um conjunto de armazenamento ZFS espelhado”](#) na página 89
- [“Colocando dispositivos off-line e on-line em um pool de armazenamento”](#) na página 92
- [“Limpando erros de dispositivo de conjunto de armazenamento”](#) na página 95
- [“Substituindo dispositivos em um pool de armazenamento”](#) na página 95
- [“Designando sobressalentes no pool de armazenamento”](#) na página 97

Adicionando dispositivos a um pool de armazenamento

É possível adicionar espaço de disco dinamicamente a um conjunto ao adicionar um novo dispositivo virtual de nível superior. Este espaço de disco fica disponível imediatamente a todos os conjuntos de dados no conjunto. Para adicionar um novo dispositivo virtual a um pool, use o comando `zpool add`. Por exemplo:

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

O formato para especificar os dispositivos virtuais é o mesmo para o comando `zpool create`. Os dispositivos são verificados para determinar se estão em uso e o comando não pode alterar o nível de redundância sem a opção `-f`. O comando também suporta a opção `-n` para que seja possível realizar uma simulação. Por exemplo:

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

Esta sintaxe de comando pode adicionar dispositivos espelhados `c3t1d0` e `c3t2d0` à configuração existente do conjunto `zeepool`.

Para obter mais informações sobre como é realizada a validação de dispositivos virtuais, consulte [“Detectando dispositivos em uso” na página 78](#).

EXEMPLO 4-1 Adicionando discos a uma configuração espelhada do ZFS

No exemplo a seguir, outro espelho é adicionado à uma configuração espelhada do ZFS em um sistema do Oracle Sun Fire x4500.

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

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

EXEMPLO 4-1 Adicionando discos a uma configuração espelhada do ZFS *(Continuação)*

```

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
```

EXEMPLO 4-2 Adicionando discos a uma configuração RAID-Z

Da mesma forma, os discos extras podem ser adicionados a uma configuração RAID-Z. O exemplo a seguir mostra como converter um conjunto de armazenamento com um dispositivo RAID-Z que contém três discos em um conjunto de armazenamento com dois dispositivos RAID-Z que contém três discos cada.

```

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

```

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

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

```

# zpool add rzpool raidz c2t2d0 c2t3d0 c2t4d0
# zpool status rzpool
pool: rzpool
state: ONLINE
scrub: none requested
config:

```

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

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

EXEMPLO 4-3 Adicionando e removendo um dispositivo de log espelhado

O exemplo abaixo mostra como adicionar um dispositivo de log espelhado a um conjunto de armazenamento espelhado. Para obter mais informações sobre a utilização de dispositivos de log no conjunto de armazenamento, consulte [“Configurando dispositivos de log do ZFS separados” na página 33.](#)

```
# 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
```

Você pode anexar um dispositivo de registro em um dispositivo existente de registro para criar um dispositivo de registro espelhado. Essa operação é idêntica à anexação de um dispositivo em um pool de armazenamento não-espelhado.

Dispositivos de log podem ser removidos utilizando o comando `zpool remove`. O dispositivo de log espelhado no exemplo anterior pode ser removido ao especificar o argumento `mirror-1`. Por exemplo:

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

EXEMPLO 4-3 Adicionando e removendo um dispositivo de log espelhado *(Continuação)*

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

Se a configuração do conjunto contém apenas um dispositivo de log, é recomendado remover o dispositivo de log ao especificar o nome do dispositivo. Por exemplo:

```
# 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
```

EXEMPLO 4-4 Adicionando e removendo dispositivos de cache

É possível adicionar ao seu conjunto de armazenamento ZFS e removê-los se não forem mais necessários.

Use o comando `zpool add` para adicionar dispositivos de cache. Por exemplo:

```
# 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

EXEMPLO 4-4 Adicionando e removendo dispositivos de cache *(Continuação)*

Os dispositivos de cache não podem ser espalhados ou fazer parte de uma configuração RAID-Z.

Use o comando `zpool remove` para remover dispositivos de cache. Por exemplo:

```
# 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
```

Atualmente, o comando `zpool remove` suporta apenas remover reservas, dispositivos de log e de cache. Os dispositivos fazem parte da configuração de pool espelhado principal e podem ser removidos usando o comando `zpool detach`. Dispositivos não-redundantes e RAID-Z não podem ser removidos de um conjunto.

Para obter mais informações sobre como utilizar dispositivos de cache em um conjunto de armazenamento ZFS, consulte [“Criando um conjunto de armazenamento do ZFS com dispositivos de cache”](#) na página 76.

Anexando e desanexando dispositivos em um pool de armazenamento

Além do comando `zpool add`, é possível utilizar o comando `zpool attach` para adicionar um novo dispositivo a um dispositivo existente espelhado ou não espelhado.

Se estiver anexando um disco para criar um conjunto raiz espelhado, consulte [“Como criar um conjunto raiz espelhado \(pós-instalação\)”](#) na página 136.

Se for substituir um disco em um conjunto raiz ZFS, consulte [“Como substituir um disco no pool raiz ZFS”](#) na página 176.

EXEMPLO 4-5 Convertendo um pool de armazenamento espelhado bidirecional em um pool de armazenamento espalhado tridirecional

Neste exemplo, `zpool` é um espelho bidirecional existente que é transformado em um espelho tridirecional ao anexar `c2t1d0`, o novo dispositivo, ao dispositivo existente `c1t1d0`.

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

    NAME          STATE      READ WRITE CKSUM
    zpool          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 zpool c1t1d0 c2t1d0
# zpool status zpool
pool: zpool
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Fri Jan  8 12:59:20 2010
config:

    NAME          STATE      READ WRITE CKSUM
    zpool          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
```

Se o dispositivo existente é parte de um espelho bidirecional, a anexação do novo dispositivo criará um espelho quadridirecional, e assim por diante. Em qualquer caso, o novo dispositivo começa a realizar `resilvering` imediatamente.

EXEMPLO 4-6 Convertendo um conjunto de armazenamento não redundante do ZFS em um conjunto de armazenamento espelhado do ZFS

Além disso, é possível converter um conjunto de armazenamento não redundante em um conjunto de armazenamento redundante utilizando o comando `zpool attach`. Por exemplo:

```
# 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
```


EXEMPLO 4-6 Convertendo um conjunto de armazenamento não redundante do ZFS em um conjunto de armazenamento espelhado do ZFS *(Continuação)*

```
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
```

O comando `zpool detach` pode ser usado para desanexar um dispositivo de um pool de armazenamento espelhado. Por exemplo:

```
# zpool detach zeepool c2t1d0
```

Entretanto, esta operação falha se nenhuma outra réplica válida dos dados existir. Por exemplo:

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

Criando um novo conjunto ao dividir um conjunto de armazenamento ZFS espelhado

Um conjunto de armazenamento ZFS espelhado pode ser clonado rapidamente como um conjunto de backup ao utilizar o comando `zpool split`.

Atualmente, este recurso não pode ser utilizado para dividir um conjunto raiz de espelhado.

É possível utilizar o comando `zpool split` para desanexar discos de um conjunto de armazenamento ZFS espelhado para criar um novo conjunto com um dos discos desanexados. O novo conjunto possuirá conteúdo idêntico ao conjunto original de armazenamento ZFS espelhado.

Por padrão, uma operação `zpool split` em um conjunto espelhado desanexa o último disco do conjunto recém criado. Depois da operação de separação, importe o novo conjunto. Por exemplo:

```
# 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
```

É possível identificar qual disco deve ser utilizado para o conjunto recém criado ao especificá-lo com o comando `zpool split`. Por exemplo:

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

Antes que a real operação de divisão ocorra, os dados na memória são descarregados para os discos espelhados. Depois que os dados são esvaziados, o disco é desanexado do conjunto e atribuído a um novo GUID de conjunto. O novo GUID de conjunto é gerado para que o conjunto possa ser importado no mesmo sistema no qual foi dividido.

Se o conjunto a ser dividido possui pontos de montagem não padrões e o novo conjunto é criado no mesmo sistema, então é necessário utilizar a opção `zpool split -R` para identificar um diretório raiz alternativo para o novo conjunto para que qualquer ponto de montagem existente não entre em conflito. Por exemplo:

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

Se não utilizar a opção `zpool split -R` e for possível visualizar que os pontos de montagem entram em conflito quando tentam importar o novo conjunto, importe o novo conjunto com a opção `-R`. Se o novo conjunto é criado em um sistema diferente, então especificar um diretório raiz alternativo não deve ser necessário a não ser que ocorra um conflito no ponto de montagem.

Revise as considerações a seguir antes de utilizar o recurso `zpool split`:

- Este recurso não está disponível para configurações RAIDZ ou conjunto não redundante de vários discos.
- Operações de aplicativo e dados devem ser pausados antes de tentar uma operação `zpool split`.
- Tendo em discos essa honra que, ao invés de ignorar, o comando de cache de gravação de esvaziamento do disco é importante.
- Um conjunto não pode ser dividido se o resilvering estiver sendo processado.
- Dividir um conjunto espelhado é o ideal quando composto de dois ou três discos, onde o último disco no conjunto original é utilizado para o conjunto recém criado. Então, é possível utilizar o comando `zpool attach` para recriar o conjunto de armazenamento original espelhado ou converter o conjunto recém criado em um conjunto de armazenamento espelhado. Não existe nenhuma maneira atualmente de criar um conjunto espelhado *novo* de um conjunto espelhado *existente* ao utilizar este recurso.
- Se o conjunto existente é um espelho tridirecional, então o novo conjunto conterà um disco depois da operação de divisão. Se o conjunto existente é um espelho bidirecional, então o resultado é de dois conjuntos não redundantes de dois discos. Será necessário anexar dois discos adicionais para converter os conjuntos não redundantes para conjuntos espelhados.
- Uma boa maneira de manter os dados redundantes durante uma operação de divisão é dividir um conjunto de armazenamento espelhado que é composto de três discos para que o conjunto original seja composto por dois discos espelhados depois da operação de divisão.

EXEMPLO 4-7 Dividindo um conjunto ZFS espelhado

No exemplo a seguir, um conjunto de armazenamento espelhado chamado `trinity`, com três discos, `c1t0d0`, `c1t2d0` e `c1t3d0`, é dividido. Os dois conjuntos resultantes são os conjuntos espelhados `trinity`, com discos `c1t0d0` e `c1t2d0`, e o novo conjunto, `neo`, com disco `c1t3d0`. Cada conjunto possui conteúdo idêntico.

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

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

EXEMPLO 4-7 Dividindo um conjunto ZFS espelhado (Continuação)

```

c1t0d0 ONLINE      0    0    0
c1t2d0 ONLINE      0    0    0
c1t3d0 ONLINE      0    0    0

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

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

errors: No known data errors

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

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

errors: No known data errors

```

Colocando dispositivos off-line e on-line em um pool de armazenamento

O ZFS permite que dispositivos sejam colocados off-line ou on-line. Quando o hardware não é confiável ou não funciona corretamente, o ZFS continua a ler ou gravar dados no dispositivo, pressupondo que a condição é apenas temporária. Se a condição não for temporária, é possível instruir o ZFS a ignorar o dispositivo colocando-o off-line. O ZFS não envia solicitações a um dispositivo off-line.

Observação – Não é necessário colocar os dispositivos off-line para substituí-los.

É possível utilizar o comando `zpool offline` quando for necessário desconectar o armazenamento temporariamente. Por exemplo, se for necessário desconectar fisicamente uma matriz de um conjunto de switches Fibre Channel e conectar a matriz a um conjunto diferente,

é possível utilizar os LUNs da matriz que foram utilizados nos conjuntos de armazenamento do ZFS. Depois da matriz ser reconectada e estar operacional no novo conjunto de switches, é possível colocar os mesmos LUNs on-line. Os dados que foram adicionados aos conjuntos de armazenamento enquanto os LUNs estavam off-line podem sofrer resilvering nos LUNs depois de serem colocados on-line.

Esta situação é possível, pressupondo que os sistemas em questão reconhecem o armazenamento depois de serem anexados aos novos switches, possivelmente através de controladores diferentes, e os conjuntos são configurados como configurações RAID-Z ou espelhados.

Colocando um dispositivo off-line

É possível colocar um dispositivo off-line com o uso do comando `zpool offline`. Se o dispositivo for um disco, ele pode ser especificado pelo caminho ou por um nome abreviado. Por exemplo:

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

Considere os pontos a seguir ao tornar um dispositivo off-line:

- Não é possível colocar um pool off-line no momento em que se torna falho. Por exemplo, não é possível colocar off-line dois dispositivos de uma configuração `raidz1`, nem colocar off-line um dispositivo virtual de nível superior.

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

- Por padrão, o estado OFF-LINE é persistente. O dispositivo permanece off-line quando o sistema é reiniciado.

Para colocar um dispositivo temporariamente off-line, use a opção `zpool offline -t`. Por exemplo:

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

Quando o sistema é reiniciado, este dispositivo volta automaticamente ao estado ONLINE.

- Quando um dispositivo é colocado off-line, ele não é desanexado do pool de armazenamento. Se tentar utilizar o dispositivo colocado off-line em outro conjunto, mesmo depois de o conjunto original ter sido destruído, será exibida uma mensagem semelhante à seguinte:

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

Se desejar utilizar o dispositivo off-line em outro conjunto de armazenamento depois de ter destruído o conjunto de armazenamento original, primeiro coloque o dispositivo online, e então destrua o conjunto de armazenamento original.

Outra forma de utilizar um dispositivo de outro conjunto de armazenamento, enquanto mantém o conjunto de armazenamento original, é substituindo o dispositivo existente no conjunto de armazenamento original por outro dispositivo semelhante. Para obter informações sobre a substituição de dispositivos, consulte [“Substituindo dispositivos em um pool de armazenamento”](#) na página 95.

Os dispositivos off-line aparecem no estado OFF - LINE quando é consultado o estado do conjunto. Para obter informações sobre consultas de status do pool, consulte [“Consultando status de pool de armazenamento do ZFS”](#) na página 106.

Para obter mais informações sobre a integridade de dispositivos, consulte [“Determinando o status de integridade de pools de armazenamento do ZFS”](#) na página 112.

Colocando um dispositivo on-line

Depois de colocar um dispositivo off-line, ele pode ser colocado novamente on-line com a utilização do comando `zpool online`. Por exemplo:

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

Quando um dispositivo é colocado on-line, quaisquer dados que tenham sido gravados no conjunto são sincronizados novamente com o dispositivo recém disponível. Observe que não é possível utilizar dispositivos on-line para substituir um disco. Se ao colocar um dispositivo off-line, substituir o dispositivo e colocá-lo on-line, ele permanece no estado de falho.

Se tentar colocar um dispositivo falho on-line, será exibida uma mensagem semelhante à seguinte:

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

Também pode ser possível visualizar a mensagem de disco falho exibida no console ou escrita no arquivo `/var/adm/messages`. Por exemplo:

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

Para obter mais informações sobre a substituição de um dispositivo falho, consulte [“Reparando um dispositivo faltando”](#) na página 299.

É possível utilizar o comando `zpool online -e` para expandir um LUN. Por padrão, um LUN que é adicionado a um conjunto não é expandido ao seu tamanho total a não ser que a propriedade do conjunto `autoexpand` esteja ativada. É possível expandir o LUN automaticamente ao utilizar o comando `zpool online -e`, mesmo se o comando LUN já estiver on-line ou se o LUN estiver atualmente off-line. Por exemplo:

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

Limpando erros de dispositivo de conjunto de armazenamento

Se um dispositivo for colocado off-line devido a uma falha que cause erros a serem listados na saída `zpool status`, você poderá limpar as contagens de erro com o comando `zpool clear`.

Se especificado sem argumentos, este comando limpará todos os erros de dispositivos dentro do pool. Por exemplo:

```
# zpool clear tank
```

Se um ou mais dispositivos forem especificados, este comando limpará somente erros associados aos dispositivos especificados. Por exemplo:

```
# zpool clear tank c1t0d0
```

Para obter mais informações sobre a limpeza de erros de `zpool`, consulte [“Apagando erros transitórios” na página 303](#).

Substituindo dispositivos em um pool de armazenamento

É possível substituir um dispositivo em um pool de armazenamento usando o comando `zpool replace`.

Se estiver substituindo fisicamente um dispositivo por outro dispositivo na mesma localização em um conjunto redundante, então só é preciso identificar o dispositivo substituído. O ZFS reconhece que o dispositivo é um disco diferente no mesmo local em alguns hardwares. Por exemplo, para substituir um disco falho (`c1t1d0`) ao remover o disco e substituindo-o na mesma localização, utilize a sintaxe semelhante à seguinte:

```
# zpool replace tank c1t1d0
```

Se estiver substituindo um dispositivo em um conjunto de armazenamento com um disco em uma localização física diferente, é necessário especificar ambos dispositivos. Por exemplo:

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

Se for substituir um disco no pool raiz ZFS, consulte [“Como substituir um disco no pool raiz ZFS” na página 176](#).

A seguir, as etapas básicas para substituir um disco:

- Coloque o disco off-line, se necessário, com o comando `zpool offline`.
- Remova o disco que será substituído.
- Insira o disco de substituição.
- Execute o comando `zpool replace`. Por exemplo:

```
# zpool replace tank c1t1d0
```

- Coloque o disco on-line outra vez com o comando `zpool online`.

Em alguns sistemas, como o Sun Fire x4500, é necessário desconfigurar o disco antes de colocá-lo off-line. Se estiver substituindo um disco na mesma posição de slot neste sistema, então é possível apenas executar o comando `zpool replace` como descrito no primeiro exemplo nesta seção.

Para um exemplo de substituição de um disco em um sistema Sun Fire X4500, consulte [Exemplo 11-1](#).

Considere o seguinte ao substituir dispositivos em um conjunto de armazenamento ZFS:

- Se a propriedade `autoreplace` do conjunto for definida como `on`, qualquer dispositivo novo encontrado na mesma localização física que um dispositivo que antes pertencia ao conjunto será automaticamente formatado e substituído. Não é necessário utilizar o comando `zpool replace` quando esta propriedade estiver ativada. Este recurso pode não estar disponível em todos os hardwares.
- O tamanho do dispositivo de substituição deve ser igual ou maior que o menor disco em uma configuração espelhada ou RAID-Z.
- Quando um dispositivo de substituição que é maior no tamanho que o dispositivo que está substituindo é adicionado ao conjunto, não é expandido automaticamente para seu tamanho total. O valor da propriedade do conjunto `autoexpand` determina se um LUN de substituição é expandido para seu tamanho total quando o disco é adicionado ao conjunto. Por padrão, a propriedade `autoexpand` é desativada. É possível ativar esta propriedade ao expandir o tamanho do LUN antes ou depois de adicionar o maior LUN ao conjunto.

No exemplo a seguir, dois discos de 16 GB em um conjunto espelhado são substituídos por dois discos de 72 GB. A propriedade `autoexpand` é ativada depois das substituições do disco para expandir os tamanho completo do LUN.


```
# 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  -
```

- Substituir muitos discos em um conjunto grande é demorado devido ao resilvering de dados nos novos discos. Além disso, deve ser considerada a execução do comando `zpool scrub` entre as substituições de disco para garantir que os dispositivos de substituição estejam operacionais e que os dados estejam sendo gravados corretamente.
- Se um disco defeituoso foi substituído automaticamente por um sobressalente, pode ser necessário desanexar o sobressalente depois que o disco defeituoso for substituído. Para obter mais informações sobre a desanexação de um sobressalente, consulte [“Ativando e desativando sobressalentes no pool de armazenamento”](#) na página 99.

Para obter mais informações sobre a substituição de dispositivos, consulte [“Reparando um dispositivo faltando”](#) na página 299 e [“Substituindo ou reparando um dispositivo modificado”](#) na página 301.

Designando sobressalentes no pool de armazenamento

O recurso de sobressalentes permite identificar os discos que poderiam ser usados para substituir um dispositivo falho ou defeituoso em um ou mais pontos de armazenamento. A atribuição de um dispositivo como *reserva* significa que o dispositivo não é um dispositivo ativo em um conjunto, mas se um dispositivo ativo no conjunto falhar, o reserva substituirá automaticamente o dispositivo falho.

Os dispositivos podem ser designados como sobressalentes nas seguintes situações:

- Quando o conjunto é criado com o comando `zpool create`.
- Após o pool ser criado com o comando `zpool add`.
- Dispositivos reservas podem ser compartilhados entre vários conjuntos, mas os sobressalentes não podem ser compartilhados entre vários conjuntos em sistemas diferentes.

Os exemplos a seguir mostram atribuir dispositivos como reservas quando o conjunto é criado:

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

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

errors: No known data errors
```

O exemplo a seguir mostra como atribuir reservas ao adicioná-los a um conjunto depois que o conjunto é criado:

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

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

errors: No known data errors
```

Os sobressalentes podem ser removidos de um pool de armazenamento com o uso do comando `zpool remove`. Por exemplo:

```
# zpool remove zeepool c2t3d0
# zpool status zeepool
pool: zeepool
```

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

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

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

Um reserva não pode ser removido se estiver sendo utilizado atualmente por um conjunto de armazenamento.

Considere o seguinte quando estiver utilizando reservas ZFS:

- Atualmente, o comando `zpool remove` pode ser utilizado apenas para remover reservas, dispositivos de cache e log.
- Para adicionar um disco como um reserva, o reserva deve ter tamanho igual ou maior ao tamanho do maior disco no conjunto. É permitido adicionar um disco menor como um sobressalente de um pool. Entretanto, quando o sobressalente menor é ativado, automaticamente ou com o comando `zpool replace`, a operação falha com um erro semelhante ao seguinte:

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

Ativando e desativando sobressalentes no pool de armazenamento

Os sobressalentes são ativados das seguintes maneiras:

- Substituição manual: substitui um dispositivo falho em um conjunto de armazenamento por um reserva utilizando o comando `zpool replace`.
- Substituição automática: quando a falha é detectada, um agente FMA examina o conjunto para verificar se existem reservas disponíveis. Se existirem, ele substituirá o dispositivo falho por um sobressalente disponível.

Se um reserva atualmente em uso falha, o agente desanexa tal sobressalente, cancelando assim a substituição. O agente tenta, então, substituir o dispositivo por outro sobressalente, se houver um disponível. Este recurso é atualmente limitado pelo fato de o mecanismo de diagnóstico do ZFS emitir falhas somente quando um dispositivo desaparece do sistema.

Se substituir fisicamente um dispositivo falho por um sobressalente ativo, é possível reativar o dispositivo original, utilizando o comando `zpool detach` para desanexar o sobressalente. Se definir a propriedade `autoreplace` do conjunto como `on`, o sobressalente é desanexado e devolvido automaticamente para o grupo disponível quando o novo dispositivo for inserido e a operação on-line terminar.

Substitua manualmente um dispositivo com um reserva utilizando o comando `zpool replace`. Consulte [Exemplo 4–8](#).

Um dispositivo falho é substituído automaticamente quando um sobressalente está disponível. Por exemplo:

```
# zpool status -x
pool: zeepool
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: resilver completed after 0h0m with 0 errors on Mon Jan 11 10:20:35 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM	
zeepool	DEGRADED	0	0	0	
mirror-0	DEGRADED	0	0	0	
c1t2d0	ONLINE	0	0	0	
spare-1	DEGRADED	0	0	0	
c2t1d0	UNAVAIL	0	0	0	cannot open
c2t3d0	ONLINE	0	0	0	88.5K resilvered
spares					
c2t3d0	INUSE	currently in use			

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

Atualmente, é possível desativar um reserva das seguintes maneiras:

- Ao remover o reserva do conjunto de armazenamento.
- Ao desanexar um reserva depois que o disco falho é substituído fisicamente. Consulte [Exemplo 4–9](#).
- Ao permutar temporariamente ou permanentemente o reserva. Consulte [Exemplo 4–10](#).

EXEMPLO 4–8 Substituindo manualmente um disco com um reserva

Neste exemplo, o comando `zpool replace` é utilizado para substituir o disco `c2t1d0` com o reserva `c2t3d0`.

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

NAME	STATE	READ	WRITE	CKSUM
zeepool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c1t2d0	ONLINE	0	0	0
spare-1	ONLINE	0	0	0
c2t1d0	ONLINE	0	0	0

EXEMPLO 4-8 Substituindo manualmente um disco com um reserva (Continuação)

```

        c2t3d0 ONLINE      0    0    0 90K resilvered
spares
  c2t3d0    INUSE      currently in use

errors: No known data errors

Então, desanexe o disco c2t1d0.

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

    NAME      STATE      READ WRITE CKSUM
    zeepool   ONLINE     0    0    0
      mirror-0 ONLINE     0    0    0
        c1t2d0 ONLINE     0    0    0
        c2t3d0 ONLINE     0    0    0 90K resilvered

errors: No known data errors

```

EXEMPLO 4-9 Desanexando um reserva depois que o disco falho é substituído

Neste exemplo, o disco falho (c2t1d0) é substituído fisicamente e o ZFS é notificado ao utilizar o comando `zpool replace`.

```

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

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

errors: No known data errors

```

Então, é possível utilizar o comando `zpool detach` para retornar o reserva para o grupo disponível. Por exemplo:

```

# zpool detach zeepool c2t3d0
# zpool status zeepool

```

EXEMPLO 4-9 Desanexando um reserva depois que o disco falho é substituído *(Continuação)*

```

pool: zeepool
state: ONLINE
scrub: resilver completed with 0 errors on Wed Jan 20 10:08:44 2010
config:

```

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

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

EXEMPLO 4-10 Desanexando um disco falho e utilizando o reserva

Se deseja substituir um disco falho ao permutar temporariamente ou permanentemente em um reserva que está atualmente substituindo-o, então desanexe o disco original (falho). Se o disco falho é substituído eventualmente, então é possível adicioná-lo de volta ao conjunto de armazenamento como um sobressalente. Por exemplo:

```

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

```

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

```

errors: No known data errors
# zpool detach zeepool c2t1d0
# zpool status zeepool
pool: zeepool
state: ONLINE
scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 13:46:46 2010
config:

```

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

EXEMPLO 4-10 Desanexando um disco falho e utilizando o reserva (Continuação)

```

c1t2d0 ONLINE      0      0      0
c2t3d0 ONLINE      0      0      0 70.5M resilvered

errors: No known data errors
(Original failed disk c2t1d0 is physically replaced)
# zpool add zeepool spare c2t1d0
# zpool status zeepool
  pool: zeepool
  state: ONLINE
  scrub: resilver completed after 0h0m with 0 errors on Wed Jan 20 13:48:46 2010
  config:

    NAME      STATE      READ WRITE CKSUM
    zeepool   ONLINE     0      0      0
      mirror-0 ONLINE     0      0      0
        c1t2d0 ONLINE     0      0      0
        c2t3d0 ONLINE     0      0      0 70.5M resilvered
      spares
        c2t1d0  AVAIL

errors: No known data errors

```

Gerenciando propriedades do pool de armazenamento do ZFS

Você pode usar o comando `zpool get` para exibir as informações das propriedades do pool. Por exemplo:

```

# zpool get all mpool
NAME PROPERTY VALUE SOURCE
pool size 68G -
pool capacity 0% -
pool altroot - default
pool health ONLINE -
pool guid 601891032394735745 default
pool version 22 default
pool bootfs - default
pool delegation on default
pool autoreplace off default
pool cachefile - default
pool failmode wait default
pool listsnapshots on default
pool autoexpand off default
pool free 68.0G -
pool allocated 76.5K -

```

As propriedades do pool de armazenamento podem ser definidas com o comando `zpool set`. Por exemplo:

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

TABELA 4-1 Descrições das propriedades do pool do ZFS

Nome da propriedade	Tipo	Valor padrão	Descrição
alocados	Sequência de caracteres	N/D	Valor de somente leitura que identifica a quantidade de espaço de armazenamento dentro do conjunto que foi fisicamente alocado.
altroot	Sequência de caracteres	off	Identifique um diretório raiz alternativo. Se definido, este diretório é anteposto a quaisquer pontos de montagem dentro do pool. Esta propriedade pode ser utilizada ao examinar um conjunto desconhecido, se os pontos de montagem não forem de confiança ou em um ambiente de inicialização alternativo no qual os caminhos típicos não são válidos.
autoreplace	Booleano	off	Controla a substituição automática do dispositivo. Se for definido como <code>desativado</code> , a substituição do dispositivo deve ser iniciada pelo administrador utilizando o comando <code>zpool replace</code> . Se for definido como <code>ativado</code> , qualquer dispositivo novo encontrado no mesmo local físico que um dispositivo que antes pertencia ao conjunto será automaticamente formatado e substituído. A abreviação da propriedade é <code>replace</code> .
bootfs	Booleano	N/D	Identifica o conjunto de dados inicializável padrão do pool raiz. Esta propriedade é tipicamente definida pelos programas de instalação e atualização.
cachefile	Sequência de caracteres	N/D	Controla onde as informações de configuração do conjunto são armazenadas. Todos os pools no cache são automaticamente importados quando o sistema é reinicializado. Entretanto, os ambientes de instalação e cluster podem precisar armazenar em cache essas informações em um local diferente, de forma que os conjuntos não sejam automaticamente importados. É possível definir esta propriedade para efetuar cache das informações de configuração do conjunto em um local diferente. Esta informação pode ser importada mais tarde ao utilizar o comando <code>zpool import - c</code> . Para a maioria das configurações ZFS, esta propriedade não é utilizada.
capacity	Número	N/D	Valor somente leitura que identifica a porcentagem de espaço usado do pool. A abreviatura da propriedade é <code>cap</code> .

TABELA 4-1 Descrições das propriedades do pool do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
<code>delegation</code>	Booleano	<code>on</code>	Controla se um usuário não-privilegiado pode obter permissões de acesso definidas para o conjunto de dados. Para obter mais informações, consulte o Capítulo 9 , “Administração delegada de ZFS”.
<code>failmode</code>	Sequência	<code>wait</code>	Controla o comportamento do sistema se uma falha catastrófica do conjunto ocorrer. Esta situação é causada normalmente pela perda de conectividade com o dispositivo ou dispositivos de armazenamento subjacentes ou por uma falha de todos os dispositivos do conjunto. O comportamento de tal evento é determinado por um dos valores a seguir: <ul style="list-style-type: none"> ■ <code>espera</code>: bloqueia todos os acessos de E/S ao conjunto até que a conectividade com o dispositivo seja restaurada e os erros sejam apagados com o comando <code>zpool clear</code>. Neste estado, operações E/S para o conjunto são bloqueadas, mas operações de leitura podem ter êxito. Um conjunto é mantido no estado <code>espera</code> até que o problema do dispositivo seja resolvido. ■ <code>continuar</code>: retorna um erro EIO a quaisquer novas solicitações de E/S escritas, mas permite a leitura dos dispositivos com integridade restantes. Quaisquer solicitações escritas que ainda não foram enviadas ao disco podem ser bloqueadas. Depois de reconectar ou substituir o dispositivo, os erros devem ser apagados com o comando <code>zpool clear</code>. ■ <code>pânico</code>: imprime uma mensagem ao console e gera um despejo de memória de sistema.
<code>livre</code>	Sequência de caracteres	N/D	Valor de somente leitura que identifica o número de bloqueios dentro do conjunto que não é alocado.
<code>guid</code>	Sequência	N/D	Valor somente leitura que identifica o identificador exclusivo do pool.
<code>health</code>	Sequência	N/D	Valor somente leitura que identifica a integridade atual do pool, como ONLINE, DEGRADED, FAULTED, OFFLINE, REMOVED ou UNAVAIL. propriedade
<code>listsnapshots</code>	Sequência de caracteres	<code>on</code>	Controla de as informações do instantâneo que estão associadas com este pool são exibidas com o comando <code>zfs list</code> . Se esta propriedade estiver desabilitada, as informações do instantâneo podem ser exibidas com o comando <code>zfs list -t snapshot</code> .

TABELA 4-1 Descrições das propriedades do pool do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
size	Número	N/D	Valor somente leitura que identifica o tamanho total do pool de armazenamento.
version	Número	N/D	Identifica a versão atual em disco do pool. O método preferido de atualização de pools é com o comando <code>zpool upgrade</code> , embora esta propriedade possa ser usada quando uma versão específica seja necessária para obter a compatibilidade com versões anteriores. Esta propriedade pode ser definida com qualquer número entre 1 e a versão atual informada pelo comando <code>zpool upgrade -v</code> .

Consultando status de pool de armazenamento do ZFS

O comando `zpool list` fornece várias maneiras de solicitar informações relacionadas ao estado do conjunto. As informações disponíveis geralmente se enquadram em três categorias: informações de uso básico, estatísticas de E/S e status de integridade. Nesta seção, são abordados os três tipos de informações do pool de armazenamento.

- “Exibindo informações sobre os conjuntos de armazenamento ZFS” na página 106
- “Visualizando estatísticas E/S para conjuntos de armazenamento ZFS” na página 110
- “Determinando o status de integridade de pools de armazenamento do ZFS” na página 112

Exibindo informações sobre os conjuntos de armazenamento ZFS

O comando `zpool list` pode ser usado para exibir informações básicas sobre pools.

Listando informações sobre todos os conjuntos de armazenamento ou de um conjunto específico

Sem nenhum argumento, o comando `zpool list` exibe as seguintes informações para todos os conjuntos no sistema:

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

Esta saída de comando exibe as informações a seguir:

NAME O nome do pool.

SIZE	O tamanho total do conjunto, igual à soma do tamanho de todos os dispositivos virtuais de nível superior.
ALLOC	A quantidade de espaço físico alocado para todos os conjuntos de dados e metadados internos. Observe que esta quantidade difere da quantidade de espaço em disco relatada no nível do sistema de arquivos. Para mais informações sobre a determinação de espaço disponível no sistema de arquivos, consulte “Contabilidade de espaço em disco do ZFS” na página 62.
FREE	A quantidade de espaço não alocado no pool.
CAP (CAPACITY)	A quantidade de espaço utilizada, expressa como uma porcentagem do espaço total de disco.
HEALTH	O status de integridade atual do pool. Para obter mais informações sobre a integridade de pools, consulte “Determinando o status de integridade de pools de armazenamento do ZFS” na página 112.
ALTROOT	A raiz alternativa do conjunto, se houver alguma. Para obter mais informações sobre pools de raiz alternativa, consulte “Usando pools de raiz alternativa do ZFS” na página 286.

Também é possível reunir estatísticas de um determinado pool especificando o nome do pool. Por exemplo:

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

Listando estatísticas específicas de pool de armazenamento

As estatísticas específicas podem ser solicitadas com o uso da opção `-o`. Esta opção proporciona relatórios personalizados ou uma forma rápida de listar informações pertinentes. Para listar, por exemplo, somente o nome e o tamanho de cada pool, você usa a seguinte sintaxe:

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

Os nomes de coluna correspondem às propriedades listadas em [“Listando informações sobre todos os conjuntos de armazenamento ou de um conjunto específico” na página 106.](#)

Script de saída de pool de armazenamento do ZFS

A saída padrão do comando `zpool list` está destinada à legibilidade, e não é fácil utilizá-la como parte de um script de shell. Para ajudar em usos programáticos do comando, a opção `-H` pode ser usada para suprimir os cabeçalhos de colunas e separar campos por guias, em vez de por espaços. Por exemplo, para solicitar uma lista simples de todos os nomes de conjuntos no sistema, utilize a seguinte sintaxe:

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

Eis um outro exemplo:

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

Exibindo o histórico de comandos do pool de armazenamento ZFS

O ZFS registra automaticamente com sucesso os comandos `zfs` e `zpool` que modificam as informações de estado do pool. Estas informações podem ser exibidas usando o comando `zpool history`.

Por exemplo, a seguinte sintaxe exhibe a saída do comando para o pool raiz:

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

É possível utilizar uma saída similar em seu sistema para identificar os comandos *exatos* do ZFS que foram executados para solucionar um problema de condição de erro.

Os recursos do registro de histórico são:

- O registro não pode ser desativado.

- O log é salvo constantemente no disco, o que significa que o log é salvo nas através das reinicializações do sistema.
- O registro é implementado como buffer de anel. O tamanho mínimo é de 128 KB. O tamanho máximo é de 32 MB.
- Para conjuntos menores, o tamanho máximo está limitado a 1% do tamanho do conjunto, nos quais o *tamanho* é determinado no momento da criação do conjunto.
- O log não requer administração, o que significa que ajustar o tamanho ou alterar a localização do log é desnecessário.

Para identificar o histórico de comandos de uma pool de armazenamento específico, use a sintaxe semelhante à seguinte:

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

Utilize a opção `-l` para exibir um formato longo que inclui o nome de usuário, o nome do host e a região na qual a operação foi efetuada. Por exemplo:

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

Use a opção `-i` para exibir de informações dos eventos internos que podem ser usadas para fins de diagnósticos. Por exemplo:

```
# zpool history -i tank
2010-05-13.14:13:15 zpool create -f tank mirror c1t2d0 c1t23d0
2010-05-13.14:13:45 [internal pool create txg:6] pool spa 19; zfs spa 19; zpl 4;...
2010-05-13.14:21:19 zfs create tank/snaps
2010-05-13.14:22:02 [internal replay_inc_sync txg:20451] dataset = 41
2010-05-13.14:25:25 [internal snapshot txg:20480] dataset = 52
2010-05-13.14:25:25 [internal destroy_begin_sync txg:20481] dataset = 41
2010-05-13.14:25:26 [internal destroy txg:20488] dataset = 41
2010-05-13.14:25:26 [internal reservation set txg:20488] 0 dataset = 0
2010-05-14.08:10:29 zfs create tank/ws01
2010-05-14.08:10:54 [internal snapshot txg:53992] dataset = 42
2010-05-14.08:10:54 zfs snapshot tank/ws01@now
2010-05-14.08:11:04 [internal create txg:53994] dataset = 58
2010-05-14.08:11:05 zfs clone tank/ws01@now tank/ws01bugfix
```

Visualizando estatísticas E/S para conjuntos de armazenamento ZFS

Para solicitar estatísticas de E/S para um pool ou dispositivos virtuais específicos, use o comando `zpool iostat`. Semelhante ao comando `iostat`, este comando pode exibir um instantâneo estático de toda atividade de E/S, bem como estatísticas atualizadas de todos os intervalos especificados. São relatadas as seguintes estatísticas:

<code>capacidade alloc</code>	A quantidade de dados atualmente armazenados no pool ou dispositivo. Esta quantidade difere um pouco da quantidade de espaço de disco disponível para os sistemas de arquivos atuais devido a detalhes de implementação interna.
	Para obter mais informações sobre as diferenças entre espaço de conjunto e espaço de conjunto de dados, consulte “Contabilidade de espaço em disco do ZFS” na página 62.
<code>capacidade livre</code>	A quantidade de espaço de disco disponível no conjunto ou dispositivo. Como com a estatística <code>used</code> , esta quantidade difere por uma pequena margem da quantidade de espaço disponível para conjuntos de dados.
<code>operações de leitura</code>	O número de operações de E/S de leitura enviado para o pool ou dispositivo, incluindo solicitações de metadados.
<code>operações de escrita</code>	O número de operações de E/S de gravação enviado para o pool ou dispositivo.
<code>ler largura de banda</code>	A largura de banda de todas as operações de leitura (incluindo metadados), expressa como unidades por segundo.
<code>escrever largura de banda</code>	A largura de banda de todas as operações de gravação, expressa como unidades por segundo.

Listando estatísticas E/S amplas de conjunto

Sem opções, o comando `zpool iostat` exibe as estatísticas acumuladas desde a inicialização de todos os pools no sistema. Por exemplo:

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

Uma vez que estas estatísticas são acumuladas desde a inicialização, a largura de banda pode parecer baixa se o pool estiver relativamente ocioso. É possível solicitar uma visão mais precisa do uso da largura de banda atual especificando um intervalo. Por exemplo:

```
# 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
```

Neste exemplo, o comando exibe estatísticas de utilização para o conjunto `tank` a cada dois segundos até que digite `Ctrl-C`. Como alternativa, é possível especificar um parâmetro de `count` adicional, que faz com que o comando termine depois do número especificado de iterações. Por exemplo, `zpool iostat 2 3` imprimiria um resumo a cada dois segundos com três iterações, em um total de seis segundos. Se houver apenas um conjunto único, as estatísticas serão exibidas em linhas consecutivas. Se houver mais de um pool, uma linha tracejada adicional delinearà cada iteração para fornecer separação visual.

Listando estatísticas E/S do dispositivo virtual

Além das estatísticas de E/S de todos os conjuntos, o comando `zpool iostat` pode exibir estatísticas de E/S para dispositivos virtuais específicos. Este comando pode ser utilizado para identificar dispositivos anormalmente lentos, ou para observar a distribuição da E/S gerada pelo ZFS. Para solicitar um layout completo do dispositivo virtual, bem como todas as estatísticas de E/S, use o comando `zpool iostat -v`. Por exemplo:

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

Note dois pontos importantes ao visualizar estatísticas E/S para dispositivos virtuais:

- Primeiro, estatísticas e utilização do espaço de disco estão disponíveis apenas para dispositivos virtuais de nível superior. A forma como o espaço de disco é alocado entre os dispositivos virtuais de espelho e RAID-Z é específica para a implementação e não é expressa facilmente como um único número.

- Em segundo lugar, os números talvez não sejam somados com exatidão, como seria de esperar. Em especial, operações nos dispositivos espelhados e RAID-Z não serão exatamente iguais. Esta diferença é particularmente evidente logo depois que um conjunto é criado, uma vez que uma quantidade significativa de E/S é feita diretamente para os discos como parte da criação do conjunto, que não é contabilizada ao nível do espelho. Com o tempo, estes números equalizam gradualmente. Entretanto, dispositivos corrompidos, que não respondem ou estão off-line também podem afetar esta simetria.

É possível usar o mesmo conjunto de opções (intervalo e contagem) ao examinar estatísticas de dispositivos virtuais.

Determinando o status de integridade de pools de armazenamento do ZFS

O ZFS fornece um método integrado de exame da integridade de pools e dispositivos. A integridade de um dispositivo é determinada a partir do estado de todos os seus dispositivos. Estas informações de estado são exibidas com o uso do comando `zpool status`. Além disso, as possíveis falhas de dispositivo e conjunto são relatadas pelo `fmdd` e exibidas no console do sistema, e registradas no arquivo `/var/adm/messages`.

Esta seção descreve como determinar a integridade de pools e dispositivos. Este capítulo não documenta como reparar ou recuperar dados de pools sem integridade. Para mais informações sobre a solução de problemas e a recuperação de dados, consulte [Capítulo 11, “Solução de problemas e conjunto de recuperação do Oracle Solaris ZFS”](#).

Cada dispositivo pode se enquadrar em um dos seguintes estados:

ONLINE	O dispositivo ou dispositivo virtual está em funcionamento de trabalho normal. Embora alguns erros transitórios ainda possam ocorrer, o dispositivo está em perfeito funcionamento.
DEGRADED	O dispositivo virtual falhou, mas continua podendo funcionar. Este estado é muito comum quando um dispositivo de espelho ou RAID-Z perde um ou mais dispositivos constituintes. A tolerância a falhas do pool pode ficar comprometida, uma vez que uma falha subsequente em outro dispositivo pode ser irrecuperável.
FAULTED	O dispositivo ou dispositivo virtual está completamente inacessível. Este estado geralmente indica falha total do dispositivo, de forma que o ZFS está incapaz de enviar ou receber dados dele. Se um dispositivo de primeiro nível estiver neste estado, o pool estará totalmente inacessível.
OFFLINE	O dispositivo foi colocado explicitamente off-line pelo administrador.

- UNAVAIL** O dispositivo ou o dispositivo virtual não pode ser aberto. Em alguns casos, conjuntos com dispositivos UNAVAIL aparecem no modo DEGRADED. Se um dispositivo virtual de nível superior estiver UNAVAIL, nada no conjunto poderá ser acessado.
- REMOVED** O dispositivo foi removido fisicamente enquanto o sistema estava em execução. A detecção da remoção do dispositivo depende do hardware e pode não ser suportada em todas as plataformas.

A integridade de um pool é determinada a partir da integridade de todos os seus dispositivos virtuais de primeiro nível. Se todos os dispositivos virtuais estiverem ONLINE, o pool também estará ONLINE. Se algum dos dispositivos virtuais estiver DEGRADED ou UNAVAIL, o conjunto também estará DEGRADED. Se um dispositivo virtual de primeiro nível estiver FAULTED ou OFFLINE, o pool também estará FAULTED. Um conjunto no estado FAULTED está completamente inacessível. Nenhum dado poderá ser recuperado até que os dispositivos necessários sejam anexados ou reparados. Um conjunto no estado DEGRADED continua a ser executado, mas talvez não atinja o mesmo nível de redundância de dados ou throughput de dados se o conjunto estivesse on-line.

Status básico de integridade do pool de armazenamento

É possível rever rapidamente estado de integridade dos conjuntos ao utilizar o comando `zpool status` como a seguir:

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

É possível examinar um determinado conjunto ao especificar o nome de um conjunto na sintaxe do comando. Qualquer pool que não esteja no estado ONLINE deve ser investigado no caso de haver problemas potenciais, como descrito na seção a seguir.

Status de integridade detalhado

É possível solicitar um resumo mais detalhado do estado da integridade ao utilizar a opção `-v`. Por exemplo:

```
# zpool status -v tank
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: scrub completed after 0h0m with 0 errors on Wed Jan 20 15:13:59 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	DEGRADED	0	0	0

```

mirror-0 DEGRADED    0    0    0
c1t0d0  ONLINE         0    0    0
c1t1d0  UNAVAIL       0    0    0  cannot open

```

errors: No known data errors

Esta saída exibe uma descrição completa dos motivos pelos quais o pool está no estado atual, incluindo uma descrição legível do problema e um link a um artigo informativo para a obtenção de mais informações. Cada artigo informativo fornece informações atualizadas sobre a melhor forma de obter recuperação do problema atual. Utilizando as informações de configuração detalhadas, é possível determinar que dispositivo está danificado e como reparar o conjunto.

No exemplo anterior, o dispositivo falho deve ser substituído. Depois de o dispositivo ser substituído, utilize o comando `zpool online` para recolocar o dispositivo on-line. Por exemplo:

```

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

```

Se a propriedade `autoreplace` está ativada, não é necessário colocar on-line o dispositivo substituído.

Se um conjunto tiver um dispositivo off-line, a saída do comando identifica o conjunto com problema. Por exemplo:

```

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

```

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

errors: No known data errors

As colunas `READ` e `WRITE` fornecem uma contagem dos erros de E/S ocorridos com o dispositivo, enquanto a coluna `CKSUM` fornece uma contagem dos erros de soma de verificação incorrigíveis que ocorreram no dispositivo. As duas contagens de erros indicam falhas potenciais do dispositivo, e algumas ações de correção são necessárias. Se forem relatados erros diferentes de zero em um dispositivo virtual de primeiro nível, partes dos dados poderão estar inacessíveis.

O campo `errors`: identifica erros de dados conhecidos.

Na saída do exemplo anterior, o dispositivo off-line não está causando erros de dados.

Para obter mais informações sobre o diagnóstico e a correção de conjuntos e dados falhos, consulte [Capítulo 11, “Solução de problemas e conjunto de recuperação do Oracle Solaris ZFS”](#).

Migrando pools de armazenamento do ZFS

Ocasionalmente, pode ser necessário mover um conjunto de armazenamento entre sistemas. Para tal, os dispositivos de armazenamento devem ser desconectados do sistema original e reconectados ao sistema de destino. Esta tarefa pode ser realizada tornando a ligar fisicamente os cabos dos dispositivos ou utilizando dispositivos de portas múltiplas, como os dispositivos em uma SAN. O ZFS permite exportar o conjunto de uma máquina e importá-lo para o sistema de destino, mesmo que os sistemas tenham arquitetura de endian diferentes. Para obter informações sobre replicação ou migração de sistemas de arquivos entre diferentes pools de armazenamento que possam residir em diferentes máquinas, consulte [“Enviando e recebendo dados do ZFS” na página 230](#).

- [“Preparando para migração do pool de armazenamento do ZFS” na página 115](#)
- [“Exportando um pool de armazenamento do ZFS” na página 116](#)
- [“Determinando pools de armazenamento disponíveis para importação” na página 116](#)
- [“Importando conjuntos de armazenamento ZFS de diretórios alternativos” na página 118](#)
- [“Importando pools de armazenamento do ZFS” na página 119](#)
- [“Recuperando pools de armazenamento destruídos do ZFS” na página 120](#)

Preparando para migração do pool de armazenamento do ZFS

Os pools de armazenamento devem ser exportados explicitamente para indicar que estão prontos para ser migrados. Esta operação libera quaisquer dados não gravados para o disco, grava os dados para o disco indicando que a exportação foi feita e remove do sistema toda a informação sobre o conjunto.

Se o pool não for explicitamente exportado, mas em vez disso for removido manualmente do disco, será possível importar o pool resultante em outro sistema. No entanto, pode ocorrer a perda dos últimos segundos das transações de dados, e o conjunto aparecerá falho no sistema original porque os dispositivos não estão mais presentes. Por padrão, o sistema de destino não pode importar um conjunto que não tenha sido exportado explicitamente. Essa condição é necessária para impedir a importação acidental de um conjunto ativo composto por armazenamentos de rede anexados que ainda estejam sendo utilizados em outro sistema.

Exportando um pool de armazenamento do ZFS

Para exportar um pool, use o comando `zpool export`. Por exemplo:

```
# zpool export tank
```

O comando tenta desmontar quaisquer sistemas de arquivos montados dentro do pool antes de continuar. Se qualquer um dos sistemas de arquivos não for desmontado, você poderá forçar a desmontagem com o uso da opção `-f`. Por exemplo:

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

Depois que o comando é executado, o conjunto `tank` deixa de ser visível no sistema.

Se os dispositivos estiverem indisponíveis no momento da exportação, os dispositivos não poderão ser especificados como exportados corretamente. Se um destes dispositivos for posteriormente anexado a um sistema sem nenhum dos dispositivos em funcionamento, ele aparecerá como "potencialmente ativo".

Se os volumes ZFS estiverem em uso no pool, o pool não poderá ser exportado, mesmo com a opção `-f`. Para exportar um conjunto com um volume ZFS, primeiro certifique-se de que todos os consumidores do volume não estão mais ativos.

Para obter mais informações sobre volumes ZFS, consulte [“Volumes ZFS” na página 277](#).

Determinando pools de armazenamento disponíveis para importação

Depois que o conjunto é removido do sistema (através de exportação explícita ou remoção forçada dos dispositivos), é possível anexar os dispositivos ao sistema de destino. O ZFS pode manipular algumas situações onde apenas alguns dispositivos estão disponíveis, mas uma migração de conjunto com êxito depende da integridade geral dos dispositivos. Além disso, os dispositivos não têm que estar necessariamente anexados sobre o mesmo nome de dispositivo. O ZFS detecta quaisquer dispositivos movidos ou renomeados e ajusta a configuração adequadamente. Para descobrir pools disponíveis, execute o comando `zpool import` sem opções. Por exemplo:

```
# 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
```

Neste exemplo, o pool tank está disponível para ser importado no sistema de destino. Cada pool é identificado por um nome e por um identificador numérico exclusivo. Se vários conjuntos com o mesmo nome estiverem disponíveis para importação, é possível utilizar identificadores numéricos para fazer a distinção entre eles.

Da mesma forma que o comando `zpool status`, o comando `zpool import` inclui um link a um artigo informativo disponível na Web com as informações mais atualizadas em relação aos procedimentos para correção de um problema que esteja impedindo um conjunto de ser importado. Neste caso, o usuário pode forçar o pool a ser importado. No entanto, a importação de um pool que esteja atualmente em uso por outro sistema em uma rede de armazenamento pode resultar na corrupção de dados e em avisos graves, uma vez que ambos os sistemas tentam gravar no mesmo armazenamento. Se alguns dispositivos não estiverem disponíveis no conjunto, mas houver redundância disponível suficiente para fornecer um conjunto utilizável, o conjunto aparecerá no estado DEGRADED. Por exemplo:

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

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

Neste exemplo, o primeiro disco está danificado ou ausente, embora você ainda possa importar o pool devido ao fato de que os dados espelhados ainda estão acessíveis. Se houver muitos dispositivos falhos ou ausentes, o pool não poderá ser importado. Por exemplo:

```
# zpool import
pool: dozer
id: 9784486589352144634
state: FAULTED
action: The pool cannot be imported. Attach the missing
       devices and try again.
       see: http://www.sun.com/msg/ZFS-8000-6X
config:
raidz1-0     FAULTED
c1t0d0      ONLINE
c1t1d0      FAULTED
```

```

c1t2d0    ONLINE
c1t3d0    FAULTED

```

Neste exemplo, dois discos de um dispositivo virtual RAID-Z estão ausentes, o que significa que não há dados replicados suficientes para reconstruir o pool. Em alguns casos, não há dispositivos suficientes para determinar a configuração completa. Neste caso, o ZFS não pode determinar quais eram os outros dispositivos que faziam parte do conjunto, embora o ZFS relate o máximo de informações possíveis sobre a situação. Por exemplo:

```

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

```

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

Importando conjuntos de armazenamento ZFS de diretórios alternativos

Por padrão, o comando `zpool import` procura apenas dispositivos dentro do diretório `/dev/dsk`. Se existirem dispositivos em outro diretório, ou se estiver utilizando conjuntos apoiados por arquivos, será necessário utilizar a opção `-d` para pesquisar diretórios alternativos. Por exemplo:

```

# 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

```

Se houver dispositivos em vários diretórios, você pode especificar várias opções `-d`.

Importando pools de armazenamento do ZFS

Depois que um conjunto é identificado para importação, é possível importá-lo especificando o nome do conjunto ou o seu identificador numérico como um argumento para o comando `zpool import`. Por exemplo:

```
# zpool import tank
```

Se vários conjuntos disponíveis tiverem o mesmo nome, especifique o conjunto que será importando utilizando o identificador numérico. Por exemplo:

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

        dozer      ONLINE
        clt9d0     ONLINE

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

        dozer      ONLINE
        clt8d0     ONLINE
# zpool import dozer
cannot import 'dozer': more than one matching pool
import by numeric ID instead
# zpool import 6223921996155991199
```

Se o nome do pool estiver em conflito com um nome de pool existente, você poderá importar o pool com outro nome. Por exemplo:

```
# zpool import dozer zeepool
```

Este comando importa o pool exportado `dozer` usando o novo nome `zeepool`.

Se o pool não tiver sido exportado corretamente, o ZFS precisará do sinalizador `-f` para impedir que os usuários importem por acidente um pool ainda em uso em outro sistema. Por exemplo:

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

Os pools também podem ser importados em uma raiz alternativa com o uso da opção `-R`. Para obter mais informações sobre pools raiz alternativos, consulte [“Usando pools de raiz alternativa do ZFS” na página 286](#).

Recuperando pools de armazenamento destruídos do ZFS

O comando `zpool import -D` pode ser usado para recuperar um pool de armazenamento destruído. Por exemplo:

```
# 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
```

Na saída `zpool import` acima, é possível identificar o conjunto `tank` como o conjunto destruído por causa das informações de estado a seguir:

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

Para recuperar o conjunto destruído, execute o comando `zpool import -D` novamente com o conjunto a ser recuperado. Por exemplo:

```
# 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
```

Se um dos dispositivos do pool destruído estiver falho ou não estiver disponível, você poderá recuperar o pool destruído mesmo assim, ao incluir a opção `-f`. Neste cenário, importe o conjunto degradado e, em seguida, tente corrigir a falha do dispositivo. Por exemplo:

```
# zpool destroy dozer
# zpool import -D
  pool: dozer
    id: 13643595538644303788
  state: DEGRADED (DESTROYED)
 status: One or more devices could not be opened. Sufficient replicas exist for
```


the pool to continue functioning in a degraded state.
 action: Attach the missing device and online it using 'zpool online'.
 see: <http://www.sun.com/msg/ZFS-8000-2Q>
 config:

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

errors: No known data errors

```
# zpool import -Df dozer
```

```
# zpool status -x
```

```
pool: dozer
```

```
state: DEGRADED
```

```
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
```

action: Attach the missing device and online it using 'zpool online'.

see: <http://www.sun.com/msg/ZFS-8000-2Q>

```
scrub: scrub completed after 0h0m with 0 errors on Thu Jan 21 15:38:48 2010
```

config:

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

errors: No known data errors

```
# zpool online dozer c2t11d0
```

```
Bringing device c2t11d0 online
```

```
# zpool status -x
```

```
all pools are healthy
```

Atualizando pools de armazenamento do ZFS

Se os conjuntos de armazenamento do ZFS forem de uma versão anterior do Solaris, como o Solaris versão 10/09, é possível atualizar os conjuntos com o comando `zpool upgrade` para tirar proveito dos recursos de conjuntos da versão do Solaris 10. Além disso, o comando `zpool status` foi modificado para notificar você quando os pools estão executando versões mais antigas. Por exemplo:

```
# 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

```

Você pode usar a sintaxe abaixo para identificar informações adicionais sobre uma versão específica e versões suportadas:

zpool upgrade -v

This system is currently running ZFS pool version 22.

The following versions are supported:

```

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

```

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

Em seguida, pode executar o comando `zpool upgrade` para atualizar todos os pools. Por exemplo:

zpool upgrade -a

Observação – Se você atualiza seu pool para uma versão ZFS mais recente, o pool não estará acessível nos sistemas que executam versões mais antigas do ZFS.

Instalando e inicializando um sistema de arquivos raiz ZFS do Oracle Solaris

Este capítulo descreve como instalar e inicializar um sistema de arquivos ZFS do Oracle Solaris. Também é tratada a migração de um sistema de arquivos raiz UFS para um sistema de arquivos ZFS com o Oracle Solaris Live Upgrade.

Este capítulo traz as seguintes seções:

- “Instalando e inicializando um sistema de arquivos ZFS do Oracle Solaris (visão geral)” na página 126
- “Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS” na página 127
- “Instalando um sistema de arquivos raiz ZFS (Instalação inicial)” na página 130
- “Instalando um sistema de arquivos raiz ZFS (instalação de arquivo do Oracle Solaris Flash)” na página 137
- “Instalando um sistema de arquivos raiz ZFS (instalação Oracle Solaris JumpStart)” na página 139
- “Migrando um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS (Oracle Solaris Live Upgrade)” na página 143
- “Suporte do ZFS para dispositivos de intercâmbio e dump” na página 165
- “Inicializando a partir de um sistema de arquivos raiz ZFS” na página 169
- “Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 176

Para uma lista de problemas conhecidos nesta versão, consulte *Notas da versão Oracle Solaris 10 9/10*.

Para obter informações atualizadas sobre a solução de problemas, visite o seguinte site:

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

Instalando e inicializando um sistema de arquivos ZFS do Oracle Solaris (visão geral)

A partir do Solaris 10 10/08, é possível instalar e inicializar a partir de um sistema de arquivos raiz ZFS das seguintes formas:

- É possível efetuar uma instalação inicial na qual o ZFS seja selecionado como sistema de arquivos raiz.
- É possível utilizar o recurso do Oracle Solaris Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS. Além disso, é possível utilizar o Oracle Solaris Live Upgrade para efetuar as seguintes tarefas:
 - Criar um novo ambiente de inicialização em um pool raiz ZFS existente.
 - Criar um novo ambiente de inicialização em um novo pool raiz ZFS.
- É possível criar um perfil JumpStart do Oracle Solaris para instalar automaticamente um sistema com um sistema de arquivos raiz ZFS.
- Começando na versão do Solaris 10 10/09, é possível criar um perfil JumpStart para instalar automaticamente um sistema com um arquivo Flash do ZFS.

Depois que um sistema com base em SPARC ou em x86 tiver sido instalado com um sistema de arquivos raiz ZFS ou migrado para um sistema de arquivos raiz ZFS, o sistema inicializa automaticamente a partir do sistema de arquivos raiz ZFS. Para obter mais informações sobre alterações na inicialização, consulte [“Inicializando a partir de um sistema de arquivos raiz ZFS” na página 169](#).

Recursos de instalação do ZFS

São oferecidos os seguintes recursos de instalação do ZFS nesta versão do Solaris:

- Com o instalador de texto interativo do Solaris, é possível instalar um sistema de arquivos raiz UFS ou ZFS. O sistema de arquivos padrão ainda é o UFS nesta versão do Solaris. Você pode acessar a opção de instalação com o instalador de texto interativo das seguintes maneiras:
 - SPARC: utilize a sintaxe a seguir do DVD de instalação do Solaris:
`ok boot cdrom - text`
 - SPARC: utilize a sintaxe a seguir quando for inicializar a partir da rede:
`ok boot net - text`
 - x86: selecione a opção de instalação no modo de texto.
- Um perfil JumpStart personalizado oferece os recursos a seguir:
 - Você pode definir um perfil para criar um pool de armazenamento do ZFS e designar um sistema de arquivos ZFS inicializável.

- É possível configurar um perfil para identificar um arquivo Flash de um pool raiz ZFS.
- Utilizando o Oracle Solaris Live Upgrade, é possível migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS. Os comandos `lucreate` e `luactivate` foram melhorados para suportar pools e sistemas de arquivos ZFS.
- É possível configurar um pool raiz ZFS espelhado selecionando dois discos durante a instalação. Ou, é possível anexar discos adicionais depois da instalação para criar um pool raiz ZFS espelhado.
- Os dispositivos de intercâmbio e dump são criados automaticamente nos volumes ZFS do pool raiz ZFS.

São oferecidos nesta versão os seguintes recursos de instalação:

- O recurso de instalação IGU para instalação de sistemas de arquivos raiz ZFS não está disponível no momento.
- O recurso de instalação Oracle Solaris Flash para instalação de sistemas de arquivos raiz ZFS não está disponível ao selecionar a opção de instalação do Flash a partir da opção de instalação inicial. No entanto, é possível criar um perfil JumpStart para identificar um arquivo Flash de um conjunto raiz ZFS. Para mais informações, consulte [“Instalando um sistema de arquivos raiz ZFS \(instalação de arquivo do Oracle Solaris Flash\)”](#) na página 137.
- Não é possível usar o programa de atualização padrão a fim de atualizar o sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS.

Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS

Certifique que os seguintes requisitos sejam atendidos antes de tentar instalar um sistema com um sistema de arquivos raiz ZFS ou tentar migrar um sistema de arquivo raiz UFS para um sistema de arquivo raiz ZFS.

Requisitos da versão do Oracle Solaris

É possível instalar e inicializar um sistema de arquivos raiz ZFS ou migrar para tal sistema das seguintes formas:

- Instalar um sistema de arquivos raiz ZFS – Disponível a partir do Solaris 10 10/08.
- Migrar de um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS com o Oracle Solaris Live Upgrade – É necessário ter no mínimo a versão Solaris 10 10/08 instalada ou ter realizado a atualização para o Solaris 10 10/08.

Requisitos gerais do pool de armazenamento ZFS

As seções a seguir descrevem o espaço e os requisitos de configuração do conjunto raiz ZFS.

Requisitos de espaço de disco para grupos de armazenamento ZFS

O montante mínimo necessário de espaço disponível para o pool em um sistema de arquivos raiz ZFS é maior do que em um sistema de arquivos raiz UFS, porque os dispositivos de intercâmbio e dump devem ser dispositivos diferentes em um ambiente de raiz ZFS. Por padrão, os dispositivos de intercâmbio e dump são um mesmo dispositivo em sistema de arquivos raiz UFS.

Quando um sistema é instalado ou atualizado com um sistema de arquivos raiz ZFS, o tamanho da área de intercâmbio e o dispositivo de dump dependem da quantidade de memória física. A quantidade mínima de espaço disponível para o pool em um sistema de arquivos raiz ZFS inicializável depende da quantidade de memória física, do espaço em disco disponível e do número de ambientes de inicialização (BEs) que serão criados.

Revise os requisitos de espaço de disco a seguir para grupos de armazenamento ZFS:

- 768 MB é a quantidade de memória mínima necessária para instalar um sistema de arquivos raiz ZFS.
- Recomenda-se 1 GB de memória para um melhor desempenho geral do ZFS.
- É recomendado como mínimo 16 GB de espaço em disco. O espaço de disco é consumido da seguinte forma:
 - **Área de permuta e dispositivo de dump** – Os tamanhos padrão dos volumes de permuta e dump criados por programas de instalação do Solaris são:
 - **Instalação inicial do Solaris** – no novo ambiente de inicialização ZFS, o tamanho do volume de permuta padrão é calculado como metade do tamanho da memória física, geralmente na média de 512 MB até 2 GB. Você pode ajustar o tamanho da intercâmbio durante uma instalação inicial.
 - O tamanho padrão do volume de dump é calculado pelo kernel com base nas informações de `dumpadm` e no tamanho da memória física. Você pode ajustar o tamanho do dump durante uma instalação inicial.
 - **Oracle Solaris Live Upgrade** – quando um sistema de arquivos raiz UFS é migrado para um sistema de arquivos raiz ZFS, o tamanho padrão do volume de permuta do ambiente de inicialização do BE do ZFS é calculado de acordo com o tamanho do dispositivo de permuta do BE do UFS. O cálculo do tamanho padrão do volume de permuta adiciona os tamanhos de todos os dispositivos de permuta do BE do UFS e cria um volume ZFS de tal tamanho no BE do ZFS. Se nenhum dispositivo de permuta estiver definido no BE do UFS, então o tamanho padrão do volume de permuta é definido em 512 MB.
 - No BE do ZFS, o tamanho do volume de despejo padrão é configurado como metade do tamanho da memória física, entre 512 MB e 2 GB.

É possível ajustar o tamanho dos volumes de permuta e despejo com o tamanho da sua preferência, contanto que os novos tamanhos suportem as operações do sistema. Para mais informações, consulte [“Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS”](#) na página 166.

- **Ambiente de inicialização (BE):** além dos novos requisitos de espaço de permuta e despejo ou ajuste dos tamanhos dos dispositivos de permuta e despejo, um BE do ZFS que é migrado de um BE do UFS precisa de 6 GB aproximadamente. Cada BE do ZFS clonado de outro BE do ZFS não requer espaço adicional em disco, mas considere que o tamanho do BE aumentará quando os patches forem aplicados. Todos os BEs do ZFS do mesmo pool raiz usam os mesmos dispositivos de intercâmbio e dump.
- **Componentes do Solaris OS:** todos os subdiretórios do sistema de arquivos raiz que são parte da imagem do sistema operacional, com exceção de `/var`, devem estar no mesmo conjunto de dados que o sistema de arquivos raiz. Além disso, todos os componentes do Solaris OS devem residir no conjunto raiz, com exceção dos dispositivos de permuta e despejo.

Outra restrição é que o diretório ou conjunto de dados `/var` deve ser um conjunto de dados único. Por exemplo, não é possível criar um conjunto de dados descendente `/var`, como `/var/tmp`, se deseja utilizar também o Oracle Solaris Live Upgrade para migrar ou instalar patch em um BE do ZFS ou criar um arquivo Flash ZFS deste conjunto.

Por exemplo, um sistema com 12 GB de espaço em disco pode ser muito pequeno para um ambiente ZFS inicializável, porque são necessários 2 GB de espaço em disco para cada dispositivo de permuta e despejo e aproximadamente 6 GB para o BE do ZFS migrado de um BE do UFS.

Requisitos da configuração pool de armazenamento ZFS

Reveja os seguintes requisitos de configuração do pool de armazenamento ZFS:

- O conjunto que está dirigido para o conjunto raiz precisa ter uma legenda SMI. Este requisito é cumprido caso o conjunto seja criado com segmentos de disco.
- O pool deve existir tanto em uma faixa de disco quanto em faixas de disco espelhadas. Se tentar utilizar uma configuração de conjunto não suportada durante a migração do Oracle Solaris Live Upgrade, será exibida uma mensagem semelhante à seguinte:

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

Para obter uma descrição detalhada das configurações suportadas do pool raiz ZFS, consulte [“Criando um pool raiz ZFS”](#) na página 73.

- **x86:** o disco deve conter uma partição `fdisk` do Solaris. É criada uma partição `fdisk` do Solaris quando o sistema com base no x86 é instalado. Para obter mais informações sobre as partições `fdisk` do Solaris, consulte [“Guidelines for Creating an fdisk Partition”](#) no *System Administration Guide: Devices and File Systems*.
- Os discos destinados à inicialização em um pool raiz ZFS devem estar limitados a um tamanho de 1 TB em sistemas baseados em SPARC e em x86.

- A compressão pode ser habilitada no pool raiz, mas somente após o pool raiz tiver sido instalado. Não existe forma de habilitar a compressão em um pool raiz durante a instalação. O algoritmo de compressão `gzip` não é suportado em pool raiz.
- Não renomeie o conjunto raiz depois dele ter sido criado como uma instalação inicial ou depois da migração do Oracle Solaris Live Upgrade para um sistema de arquivos raiz ZFS. Renomear o conjunto raiz pode causar em um sistema não inicializável.

Instalando um sistema de arquivos raiz ZFS (Instalação inicial)

Nesta versão do Solaris, você pode realizar uma instalação inicial usando o instalador de texto interativo do Solaris para criar um pool de armazenamento do ZFS que contenha um sistema de arquivos raiz ZFS inicializável. Se tiver um grupo de armazenamento ZFS que deseja utilizar no sistema de arquivos raiz ZFS, deve ser utilizado o Oracle Solaris Live Upgrade para migrar o sistema de arquivos raiz UFS existente em um conjunto de armazenamento ZFS. Para mais informações, consulte [“Migrando um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS \(Oracle Solaris Live Upgrade\)”](#) na página 143.

Se for configurar regiões depois da instalação inicial de um sistema de arquivos raiz ZFS e tiver planos de atualizar ou instalar patches no sistema, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(Solaris 10 10/08\)”](#) na página 149 ou [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(no mínimo Solaris 10 5/09\)”](#) na página 155.

Se já possuir conjuntos de armazenamento ZFS no sistema, eles são reconhecidos pela mensagem a seguir. Entretanto, estes conjuntos permanecem intocados, a menos que selecione os discos nos conjuntos existentes para criar um novo conjunto de armazenamento.

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



Cuidado – Os pools existentes serão destruídos se qualquer um dos seus discos estiverem selecionados no novo pool.

Antes de começar a instalação inicial para criar um conjunto de armazenamento ZFS, consulte [“Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS”](#) na página 127.

EXEMPLO 5-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável

O processo de instalação de texto interativo do Solaris é basicamente o mesmo das versões anteriores do Solaris, exceto que você é solicitado a criar um sistema de arquivos raiz UFS ou ZFS. O UFS ainda é o sistema de arquivos padrão nesta versão. Se selecionar um sistema de

EXEMPLO 5-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável *(Continuação)*

arquivos raiz ZFS, será solicitada a criação de um conjunto de armazenamento ZFS. As etapas para instalar o sistema de arquivos raiz ZFS seguem:

1. Selecione o método de instalação interativo do Solaris porque a instalação com o Solaris Flash não está disponível para criar um sistema de arquivos raiz ZFS inicializável. Entretanto, é possível criar um arquivo Flash ZFS para ser utilizado durante a instalação do JumpStart. Para mais informações, consulte [“Instalando um sistema de arquivos raiz ZFS \(instalação de arquivo do Oracle Solaris Flash\)”](#) na página 137.

Começando com a versão do Solaris 10 10/08, é possível migrar de um sistema de arquivo raiz UFS para um sistema de arquivos raiz ZFS desde que a versão do Solaris 10 10/08 ou posterior já esteja instalada. Para mais informações sobre a migração para um sistema de arquivos raiz ZFS, consulte [“Migrando um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS \(Oracle Solaris Live Upgrade\)”](#) na página 143.

2. Para criar um sistema de arquivos raiz ZFS, selecione a opção ZFS. Por exemplo:

Choose Filesystem Type

Select the filesystem to use for your Solaris installation

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

3. Após selecionar o software que será instalado, você é solicitado a selecionar os discos para criar o pool de armazenamento ZFS. Esta tela é semelhante à das versões anteriores do Solaris.

Select Disks

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

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

Maximum Root Size: 69994 MB
Suggested Minimum: 8279 MB

Você pode selecionar o disco ou os discos a serem usados para o pool raiz ZFS. Se selecionar dois discos, uma configuração espelhada de dois discos é definida para o pool raiz. Tanto um conjunto espelhado de dois discos quanto de três discos é excelente. Se tiver oito discos e selecionar todos eles, os oito discos são utilizados para o conjunto raiz como um grande

EXEMPLO 5-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável *(Continuação)*

espelho. Esta configuração não é considerada ótima. Outra opção é criar um pool raiz espalhado após a conclusão da instalação inicial. Uma configuração de pool RAID-Z para o pool raiz não é suportada. Para obter mais informações sobre a configuração de pools de armazenamento do ZFS, consulte [“Recursos de replicação de um pool de armazenamento do ZFS” na página 69](#).

4. Para selecionar dois discos para criar um conjunto raiz espelhado, utilize as teclas de controle do cursor para selecionar o segundo disco. Por exemplo, ambos `c1t1d0` e `c0t2d0` são selecionados para os discos de conjunto raiz. Ambos os discos precisam ter uma legenda SMI e uma fatia 0. Se os discos não estão legendados com um SMI ou não contém segmentos, é preciso sair do programa de instalação. Utilize o utilitário `format` para legendar novamente e reparticionar os discos e, a seguir, reinicie o programa de instalação.

Select Disks

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

Disk Device	Available Space
[X] c1t0d0	69994 MB
[X] c1t1d0	69994 MB (F4 to edit)
[-] c1t2d0	0 MB
[-] c1t3d0	0 MB
Maximum Root Size: 69994 MB	
Suggested Minimum: 8279 MB	

Se a coluna Espaço disponível identifica 0 MB, o disco provavelmente possui uma legenda EFI. Se deseja utilizar um disco com uma legenda EFI, é necessário sair da instalação do programa, relegendar o disco com uma legenda SMI ao utilizar o comando `format -e e`, então, reinicializar o programa de instalação.

Se não for criado um conjunto raiz espelhado durante a instalação, é possível criar um facilmente depois da instalação. Para informações, consulte [“Como criar um conjunto raiz espelhado \(pós-instalação\)” na página 136](#).

5. Depois de selecionar um ou mais discos para o conjunto de armazenamento ZFS, será exibida uma tela semelhante à seguinte:

Configure ZFS Settings

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

ZFS Pool Name: rpool

EXEMPLO 5-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável *(Continuação)*

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

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

Nesta tela, é possível alterar o nome do conjunto ZFS, o nome do conjunto de dados, o tamanho do conjunto e o tamanho dos dispositivos de permuta e despejo movendo as teclas de controle do cursor pelas entradas e substituindo o valor do texto padrão pelo novo valor. Ou, pode aceitar os valores padrão. Além disso, é possível modificar a forma como o sistema de arquivos /var é criado e montado.

Neste exemplo, o nome do conjunto de dados é alterado para zfsBE.

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

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

6. Você pode alterar o perfil da instalação nesta tela final de instalação. Por exemplo:

Profile

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

```
=====

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

Regions: North America
System Locale: C ( C )

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

7. Quando a instalação estiver concluída, revise as informações resultantes do conjunto de armazenamento e do sistema de arquivos ZFS. Por exemplo:

```
# zpool status
pool: rpool
```

EXEMPLO 5-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável *(Continuação)*

```

state: ONLINE
scrub: none requested
config:

NAME          STATE      READ WRITE CKSUM
rpool         ONLINE    0     0     0
  mirror-0    ONLINE    0     0     0
    clt0d0s0  ONLINE    0     0     0
    clt1d0s0  ONLINE    0     0     0

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

```

A saída `zfs list` de amostra identifica os componentes do pool raiz, como o diretório `rpool/ROOT`, que não está acessível por padrão.

- Para criar outro ambiente de inicialização (BE) ZFS no mesmo conjunto de armazenamento, é possível utilizar o comando `lucreate`. No exemplo seguinte, é criado um novo BE denominado `zfs2BE`. O BE atual é nomeado `zfsBE`, como mostrado na saída `zfs list`. Entretanto, o BE atual não é reconhecido na saída `lustatus` até que o novo BE seja criado.

```

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

```

Se criar um novo BE do ZFS no mesmo pool, use uma sintaxe semelhante à seguinte:

```

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

```

EXEMPLO 5-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

Criando um BE do ZFS dentro do mesmo conjunto utiliza recursos de clone ZFS e instantâneo para criar instantaneamente o BE. Para mais detalhes sobre a utilização do Oracle Solaris Live Upgrade em uma migração de raiz ZFS, consulte [“Migrando um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS \(Oracle Solaris Live Upgrade\)”](#) na página 143.

- Depois, verifique os novos ambientes de inicialização. Por exemplo:

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

# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                8.03G 58.9G   97K    /rpool
rpool/ROOT           4.47G 58.9G   21K    legacy
rpool/ROOT/zfs2BE    116K  58.9G  4.47G   /
rpool/ROOT/zfsBE     4.47G 58.9G  4.47G   /
rpool/ROOT/zfsBE@zfs2BE 75.5K -      4.47G   -
rpool/dump           1.50G 58.9G  1.50G   -
rpool/export         44K   58.9G  23K    /export
rpool/export/home    21K   58.9G  21K    /export/home
rpool/swap           2.06G 61.0G  16K    -
```

- Se desejar inicializar a partir de BE alternativo, utilize o comando `luactivate`. Depois de ativar o BE em um sistema com base em SPARC, utilize o comando `boot - L` para identificar os BEs disponíveis quando o dispositivo de inicialização contiver um conjunto de armazenamento ZFS. Ao inicializar de um sistema com base no x86, identifique o BE a ser inicializado no menu GRUB.

Por exemplo, em um sistema com base no SPARC, use o comando `boot - L` para exibir uma lista dos BEs disponíveis. Para inicializar do novo BE, `zfs2BE`, selecione a opção 2. Em seguida, digite o comando `boot - Z` exibido.

```
ok boot -L
Executing last command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0 File and args: -L
1 zfsBE
2 zfs2BE
Select environment to boot: [ 1 - 2 ]: 2

To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfs2BE
ok boot -Z rpool/ROOT/zfs2BE
```

Para obter mais informações sobre a inicialização de um sistema de arquivos ZFS, consulte [“Inicializando a partir de um sistema de arquivos raiz ZFS”](#) na página 169.

▼ Como criar um conjunto raiz espelhado (pós-instalação)

Se não for criado um conjunto raiz espelhado durante a instalação, é possível criar um facilmente depois da instalação.

Para informações sobre a substituição de um disco no conjunto raiz ZFS, consulte [“Como substituir um disco no pool raiz ZFS” na página 176](#).

1 Exibe o estado atual do conjunto raiz.

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

NAME        STATE      READ WRITE CKSUM
rpool       ONLINE    0     0     0
  ct10d0s0  ONLINE    0     0     0

errors: No known data errors
```

2 Anexe o segundo disco para configurar um conjunto raiz espelhado.

```
# zpool attach rpool ct10d0s0 ct11d0s0
Please be sure to invoke installboot(1M) to make 'ct11d0s0' bootable.
Make sure to wait until resilver is done before rebooting.
```

3 Visualize o estado do conjunto raiz para confirmar que o resilvering foi concluído.

```
# zpool status rpool
pool: rpool
state: ONLINE
status: One or more devices is currently being resilvered. The pool will
continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scrub: resilver in progress for 0h1m, 24.26% done, 0h3m to go
config:

NAME        STATE      READ WRITE CKSUM
rpool       ONLINE    0     0     0
  mirror-0  ONLINE    0     0     0
    ct10d0s0 ONLINE    0     0     0
    ct11d0s0 ONLINE    0     0     0  3.18G resilvered

errors: No known data errors
```

Na saída acima, o processo de resilvering não está concluído. Resilvering está concluído quando é possível visualizar mensagens semelhantes à seguinte:

```
scrub: resilver completed after 0h10m with 0 errors on Thu Mar 11 11:27:22 2010
```


4 Aplique bloqueios de inicialização ao segundo disco depois da conclusão do resilvering.

```
sparc# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t1d0s0
```

```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

5 Verifique se é possível inicializar com êxito a partir do segundo disco.**6 Configure o sistema para inicializar automaticamente a partir do disco novo, ou utilizando o comando eeprom, o comando setenv da PROM de inicialização do SPARC. Ou reconfigure a BIOS do PC.**

Instalando um sistema de arquivos raiz ZFS (instalação de arquivo do Oracle Solaris Flash)

Iniciando na versão do Solaris 10 10/09, é possível criar um arquivo Flash em um sistema que executa um sistema de arquivos raiz UFS ou ZFS. O arquivo Flash de um sistema de arquivos raiz ZFS contém toda a hierarquia do pool, exceto para volumes de dump e intercâmbio, e todos os conjuntos de dados excluídos. Os volumes de permuta e despejo são criados quando o arquivo Flash é instalado. É possível utilizar o método de instalação do arquivo Flash da seguinte forma:

- Gere um arquivo Flash que possa ser utilizado para instalar e inicializar um sistema com um sistema de arquivos raiz ZFS.
- Efetue uma instalação JumpStart de um sistema utilizando um arquivo Flash ZFS. A criação de um arquivo Flash ZFS clona todo um conjunto raiz, não dos ambientes de inicialização individuais. É possível excluir os conjuntos de dados individuais dentro do grupo utilizando o comando `flarcreate` e a opção `D` do comando `-flar`.

Revise as seguintes limitações antes de considerar a instalação de um sistema com um arquivo Flash ZFS:

- Apenas uma instalação JumpStart de um arquivo Flash ZFS é suportada. Você não pode usar a opção de instalação interativa de um arquivo Flash para instalar um sistema com um sistema de arquivos raiz ZFS. Não é possível utilizar um arquivo Flash para instalar um BE do ZFS com o Oracle Solaris Live Upgrade.
- Só é possível instalar um arquivo Flash em um sistema que possui a mesma arquitetura que o sistema no qual foi criado o arquivo Flash ZFS. Por exemplo, um arquivo que é criado em um sistema `sun4u` não pode ser instalado em um sistema `sun4v`.
- Apenas uma instalação inicial completa de um arquivo Flash ZFS é suportada. Não é possível instalar um arquivo Flash diferencial de um sistema de arquivos raiz ZFS nem é possível instalar um arquivo UFS/ZFS híbrido.

- Os arquivos Flash do UFS existentes ainda podem ser utilizados para instalar um sistema de arquivos raiz UFS. O arquivo Flash do ZFS somente pode ser usado para instalar um sistema de arquivos raiz do ZFS.
- Embora todo o conjunto raiz, menos quaisquer conjuntos de dados explicitamente excluídos, é arquivado e instalado, apenas o BE do ZFS que é inicializado quando o arquivo é criado é utilizável depois da instalação do arquivo Flash. Entretanto, os conjuntos que são arquivados com o comando `flar` ou `flarcreate` com a opção `-R` do comando, a opção `rootdir` pode ser utilizadas para arquivar o conjunto raiz que não seja o que está sendo inicializado no momento.
- Um nome de conjunto raiz ZFS que é criado com um arquivo Flash precisa corresponder ao nome do conjunto raiz mestre. O nome do conjunto raiz que é utilizado para criar o arquivo Flash é o nome que está atribuído ao novo grupo criado. A alteração do nome do pool não é suportada.
- As opções do comando `flarcreate` e `flar` para incluir e excluir arquivos individuais não são suportadas em um arquivo Flash ZFS. É possível excluir apenas o conjunto completo de dados de um arquivo Flash ZFS.
- O comando `flar info` não é suportado para um arquivo Flash ZFS. Por exemplo:

```
# flar info -l zfs10u8flar
ERROR: archive content listing not supported for zfs archives.
```

Depois que um sistema mestre é instalado ou atualizado para a versão do Solaris 10 10/09, é possível criar um arquivo Flash ZFS para ser utilizado para instalar um sistema alvo. O processo básico é:

- Instale ou atualize para pelo menos a versão do Solaris 10 10/09 no sistema mestre. Adicione quaisquer personalizações que desejar.
- Crie o arquivo Flash ZFS com o comando `flarcreate` no sistema mestre. Todos os conjuntos de dados no conjunto raiz, exceto para os volumes de permuta e despejo, são incluídos no arquivo Flash ZFS.
- Crie um perfil JumpStart para incluir as informações do arquivo flash no servidor de instalação.
- Instale o arquivo Flash ZFS no sistema de destino.

As opções de arquivamento a seguir são suportadas para a instalação do conjunto raiz ZFS com um arquivo Flash:

- Utilize o comando `flarcreate` ou `flar` para criar um arquivo Flash do conjunto raiz ZFS especificado. Se não for especificado, um arquivo Flash do conjunto raiz padrão é criado.
- Utilize `flarcreate -D dataset` para excluir os conjuntos de dados especificados do arquivo Flash. Esta opção pode ser usada múltiplas vezes para excluir múltiplos conjuntos de dados.

Após a instalação de uma arquivo Flash ZFS, o sistema é configurado como segue:

- Toda a hierarquia do conjunto de dados que existia no sistema onde o arquivo Flash foi criado é recriada no sistema destino, menos quaisquer conjuntos de dados que foram especificamente excluídos no momento da criação do arquivo. Os volumes de permuta e despejo não estão incluídos no arquivo Flash.
- O pool raiz tem o mesmo nome do pool que foi usado para criar o arquivo.
- O ambiente de inicialização que estava ativo quando o arquivo Flash foi criado é o BE ativo e padrão nos sistemas implementados.

EXEMPLO 5-2 Instalando um sistema com um arquivo Flash do ZFS

Depois da instalação ou atualização de um sistema mestre com no mínimo a versão do Solaris 10 10/09, crie um arquivo Flash para o conjunto raiz ZFS. Por exemplo:

```
# flarcreate -n zfsBE zfs10upflar
Full Flash
Checking integrity...
Integrity OK.
Running precreation scripts...
Precreation scripts done.
Determining the size of the archive...
The archive will be approximately 4.94GB.
Creating the archive..
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.

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

No sistema que será utilizado como o servidor de instalação, crie um perfil JumpStart da mesma maneira que faria para instalar qualquer sistema. Por exemplo, o perfil a seguir é utilizado para instalar o arquivo `zfs10flar`.

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

Instalando um sistema de arquivos raiz ZFS (instalação Oracle Solaris JumpStart)

É possível criar um perfil JumpStart para instalar um sistema de arquivos raiz ZFS ou um sistema de arquivos raiz UFS.

Um perfil específico do ZFS deve conter a nova palavra-chave `pool`. A palavra-chave `conjunto` instala um novo conjunto raiz e, por padrão, é criado um novo ambiente de inicialização. É possível fornecer o nome do ambiente de inicialização e criar outro conjunto de dados em separado `/var` com as palavras-chave `bootenv installbe` e as opções `bename` e `dataset`.

Para obter informações gerais sobre o uso dos recursos do JumpStart, consulte [Guia de instalação do Oracle Solaris 10 9/10: instalação JumpStart personalizada e instalações avançadas](#).

Se for configurar regiões depois da instalação JumpStart de um sistema de arquivos raiz ZFS e tiver planos de atualizar ou instalar patches no sistema, consulte “[Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(Solaris 10 10/08\)](#)” na página 149 ou “[Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(no mínimo Solaris 10 5/09\)](#)” na página 155.

Palavras-chave do JumpStart para ZFS

Em um perfil específico do ZFS são permitidas as seguintes palavras-chave:

auto Especifica automaticamente o tamanho dos segmentos do grupo, do volume de permuta ou do volume de despejo. O tamanho do disco é verificado para comprovar que o tamanho mínimo pode ser alojado. Se o tamanho mínimo puder ser acomodado, o maior tamanho possível do conjunto é alocado, dadas as limitações, como o tamanho dos discos, os segmentos preservados, e assim por diante.

Por exemplo, se especificar `c0t0d0s0`, o segmento de conjunto raiz é criado com o maior tamanho possível tanto se especificar a palavra-chave `all` quanto `auto`. Ou é possível especificar um tamanho determinado para o segmento ou para o volume de permuta ou despejo.

A palavra-chave `auto` funciona de forma semelhante à palavra-chave `all` quando utilizada com um conjunto raiz ZFS porque os conjuntos não têm espaço não utilizado.

bootenv Identifica as características do ambiente de inicialização.

Use a seguinte sintaxe da palavra-chave `bootenv` para criar um ambiente de raiz ZFS inicializável:

```
bootenv installbe bename BE-name [ dataset mount-point ]
```

`installbe` Cria um novo BE identificado pela opção `bename` e pela entrada `nome-do-BE` e o instala.

`bename nome-do-BE` Identifica o `nome-do-BE` a ser instalado.

Se `bename` não for usado com a palavra-chave `pool`, é criado um BE padrão.

`dataset mount-point` Use a palavra-chave `dataset` opcional para identificar o conjunto de dados `/var`, que é diferente do conjunto de

dados raiz. O valor do *mount-point* está atualmente limitado a */var*. Por exemplo, a linha de sintaxe de *bootenv* de um conjunto de dados */var* diferente é semelhante à seguinte:

```
bootenv installbe bename zfsroot dataset /var
```

pool Define o novo pool raiz que será criado. Deve ser fornecida a seguinte sintaxe da palavra-chave:

```
pool poolname poolsize swapsize dumpsize vdevlist
```

poolname Identifica o nome do pool que será criado. O pool é criado com o *tamanho* especificado e com os dispositivos físicos especificados (*vdevs*). O valor *poolname* não deve identificar o nome de um conjunto existente ou o conjunto existente será substituído.

poolsize Especifica o tamanho do pool que será criado. O valor pode ser *auto* ou *existing*. O valor *auto* aloca o maior tamanho possível do conjunto, dadas as limitações, como o tamanho dos discos, os segmentos preservados e assim por diante. O valor *existing* significa que os limites dos segmentos existentes com tal nome são preservados ou substituídos. O tamanho adotado é em MB, a menos que esteja especificado por *g* (GB).

swapsize Especifica o tamanho do volume de intercâmbio que será criado. O valor *auto* significa que o tamanho de permuta padrão é utilizado. É possível especificar um tamanho com um valor *size*. O tamanho adotado é em MB, a menos que estiver especificado por *g* (GB).

dumpsize Especifica o tamanho do volume de dump que será criado. O valor *auto* significa que o tamanho de permuta padrão é utilizado. É possível especificar um tamanho com um valor *size*. O tamanho presumido é em MB, a menos que esteja especificado por *g* (GB).

vdevlist Especifica um ou mais dispositivos que serão usados para criar o pool. O formato de *vdevlist* é o mesmo que o formato do comando *zpool create*. Atualmente, somente as configurações espelhadas são suportadas quando vários dispositivos são especificados. Os dispositivos de *vdevlist* devem ser segmentos para o conjunto raiz. O valor *any* significa que o software de instalação seleciona um dispositivo adequado.

É possível espelhar quantos discos quiser, mas o tamanho do conjunto criado é determinado pelo menor dos discos especificados. Para obter mais informações sobre a criação de pools de armazenamento espelhados, consulte [“Configuração de pool de armazenamento espelhado”](#) na página 69.

Exemplos de perfil do JumpStart para ZFS

Esta seção oferece exemplos de perfis do JumpStart específicos do ZFS.

O perfil seguinte realiza uma instalação inicial especificada com `install_type initial_install` em um novo pool, identificado com pool `newpool`, cujo tamanho é automaticamente determinado com a palavra-chave `auto` para o tamanho dos discos especificados. A área de permuta e dispositivo de despejo é automaticamente dimensionada com a palavra-chave `auto` em uma configuração espelhada dos discos (com a palavra-chave `mirror` e discos especificados como `c0t0d0s0` e `c0t1d0s0`). As características do ambiente de inicialização são definidas com a palavra-chave `bootenv` para instalar um novo BE com a palavra-chave `installbe` e um bename denominado `s10-xx` é criado.

```
install_type initial_install
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename s10-xx
```

O perfil a seguir efetua uma instalação inicial com a palavra-chave `install_type initial_install` do metacluster `SUNWCall` em um novo grupo chamado `newpool`, que possui tamanho de 80 GB. Este conjunto é criado com um volume de permuta de 2 GB e um volume de despejo de 2 GB, em uma configuração espelhada de qualquer um dos dois dispositivos disponíveis que seja grande o suficiente para criar um grupo de 80 GB. Se nenhum dos dois dispositivos estiverem disponíveis, a instalação falha. As características do ambiente de inicialização são definidas com a palavra-chave `bootenv` para instalar um novo BE com a palavra-chave `installbe` e um bename denominado `s10-xx` é criado.

```
install_type initial_install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename s10-xx
```

A sintaxe de instalação do JumpStart permite a preservação ou criação de um sistema de arquivos UFS em um disco que inclui também o conjunto raiz ZFS. Esta configuração não é recomendada para sistemas de produção, mas pode ser usada para transições ou migrações em sistemas pequenos, tal como um laptop.

Problemas do JumpStart ZFS

Considere os problemas a seguir antes de iniciar uma instalação JumpStart de um sistema de arquivos raiz ZFS inicializável.

- Não é possível usar um pool de armazenamento do ZFS existente em uma instalação JumpStart para criar um sistema de arquivos raiz ZFS inicializável. Você deve criar um novo pool de armazenamento do ZFS com uma sintaxe semelhante à seguinte:

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

- É necessário criar o conjunto com segmentos de disco e não com o disco inteiro, conforme descrito em [“Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS” na página 127](#). Por exemplo, a sintaxe em negrito no exemplo a seguir não é aceitável:

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

A sintaxe em negrito no exemplo a seguir é aceitável:

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

Migrando um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS (Oracle Solaris Live Upgrade)

Os recursos relacionados do Oracle Solaris Live Upgrade estão disponíveis e, se estiverem relacionados aos componentes UFS, funcionam como nas versões anteriores do Solaris.

Os recursos a seguir também estão disponíveis:

- Ao migrar o sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS, você deve designar um pool de armazenamento do ZFS existente com a opção `-p`.
- Se o sistema de arquivos raiz UFS tiver componentes em diferentes faixas, eles são migrados para o pool raiz ZFS.
- É possível migrar um sistema com regiões, mas as configurações suportadas são limitadas no Solaris 10 10/08. Mais configurações de regiões são suportadas a partir da versão do Solaris 10 5/09. Para obter mais informações, consulte as seguintes seções:
 - [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(Solaris 10 10/08\)” na página 149](#)
 - [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(no mínimo Solaris 10 5/09\)” na página 155](#)

Se for migrar para um sistema de arquivos raiz ZFS sem regiões, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar para um sistema de arquivos raiz ZFS \(sem regiões\)” na página 145](#).

- O Oracle Solaris Live Upgrade pode utilizar os recursos de clone e instantâneo ZFS ao criar um BE do ZFS no mesmo conjunto. Conseqüentemente, a criação do BE é muito mais rápida do que nas versões anteriores do Solaris.

Para informações detalhadas sobre a instalação do Oracle Solaris e os recursos do Oracle Solaris Live Upgrade, consulte [Oracle Guia de instalação do Solaris 10 9/10: Solaris Live Upgrade e planejamento da atualização](#).

O processo básico para migrar de um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS é o seguinte:

- Instale Solaris 10 10/08, Solaris 10 5/09, Solaris 10 10/09 ou o Solaris 10 9/10, ou utilize o programa de atualização padrão para atualizar de uma versão anterior do Solaris 10 em qualquer sistema suportado com base em SPARC ou x 86.
- Quando estiver executando o Solaris 10 10/08, crie um conjunto de armazenamento ZFS para o sistema de arquivos raiz ZFS.
- Utilize o Oracle Solaris Live Upgrade para migrar o sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS.
- Ative o BE do ZFS com o comando `luactivate`.

Para informações sobre requisitos do ZFS e do Oracle Solaris Live Upgrade, consulte [“Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS”](#) na página 127.

Problemas de migração ZFS com o Oracle Solaris Live Upgrade

Revise a lista de problemas a seguir antes de utilizar o Oracle Solaris Live Upgrade para migrar o sistema de arquivos raiz UFS para um ZFS:

- A opção de atualização padrão da GUI de instalação do Oracle Solaris não está disponível para migração de um sistema de arquivos raiz UFS para ZFS. Para migrar de um sistema de arquivos UFS, você deve utilizar o Oracle Solaris Live Upgrade.
- É necessário criar o conjunto de armazenamento ZFS que será utilizado para a inicialização antes da operação com o Oracle Solaris Live Upgrade. Além disso, devido às limitações atuais da inicialização, o pool raiz ZFS deve ser criado com faixas em vez de discos inteiros. Por exemplo:

```
# zpool create rpool mirror c1t0d0s0 c1t1d0s0
```

Antes de criar o novo conjunto, certifique-se de que os discos a serem utilizados no grupo tenham uma legenda SMI (VTOC) em vez de uma legenda EFI. Se o disco for remarcado com uma legenda SMI, certifique-se de que o processo de rotulação não altera o esquema de particionamento. Na maioria dos casos, toda capacidade do disco deve estar nos segmentos destinados ao conjunto raiz.

- Não é possível utilizar o Oracle Solaris Live Upgrade para criar um BE do UFS a partir de um BE do ZFS. Se migrar o BE do UFS para um BE do ZFS e reter o BE do UFS, você pode inicializar tanto do BE do UFS quanto do BE do ZFS.
- Não renomeie os BEs do ZFS com o comando `zfs rename`, porque o recurso do Oracle Solaris Live Upgrade não pode detectar a alteração de nome. Os comando subseqüentes, como `ludelete`, falharão. De fato, não renomeie os pools ou sistemas de arquivos ZFS se tiver BEs que deseja continuar usando.
- Ao criar um BE alternativo que é um clone do BE principal, você não pode usar as opções `-f`, `-x`, `-y`, `-Y` e `-z` para incluir ou excluir arquivos de tal BE principal. É possível usar as opções de inclusão e exclusão definidas nos seguintes casos:

```
UFS -> UFS
UFS -> ZFS
ZFS -> ZFS (different pool)
```

- Embora seja possível utilizar o Oracle Solaris Live Upgrade para atualizar um sistema de arquivos raiz UFS para um ZFS, não é possível utilizá-lo para atualizar sistemas de arquivos não raiz ou compartilhados.
- Não é possível usar o comando `lu` para criar e migrar um sistema de arquivos raiz ZFS.

Utilizando o Oracle Solaris Live Upgrade para migrar para um sistema de arquivos raiz ZFS (sem regiões)

Os exemplos seguinte mostram como migrar um sistema de arquivos raiz UFS para ZFS.

Se for migrar ou atualizar um sistema com regiões, consulte as seguintes seções:

- [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(Solaris 10 10/08\)”](#) na página 149
- [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(no mínimo Solaris 10 5/09\)”](#) na página 155

EXEMPLO 5-3 Utilizando o Oracle Solaris Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS

O exemplo seguinte mostra como criar um BE de um sistema de arquivos raiz ZFS a partir de um sistema de arquivos raiz UFS. O BE atual, `ufsBE`, que contém um sistema de arquivos raiz UFS, é identificado pela opção `-c`. Se a opção `-c` não for incluída, o nome do BE atual será, por padrão, o nome do dispositivo. O novo BE, `zfsBE`, é identificado pela opção `-n`. Um pool de armazenamento ZFS deve existir antes da operação com `lucreate`.

O conjunto de armazenamento ZFS deve ser criado com segmentos em vez de com um disco inteiro para que possa ser atualizável e inicializável. Antes de criar o novo grupo, certifique-se de que os discos a serem utilizados no grupo tenham uma legenda SMI (VTOC) em vez de uma legenda EFI. Se o disco for remarcado com uma legenda SMI, certifique-se de que o processo de

EXEMPLO 5-3 Utilizando o Oracle Solaris Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS (Continuação)

rotulação não altera o esquema de particionamento. Na maioria dos casos, toda capacidade do disco deve estar nos segmentos destinados ao conjunto raiz.

```
# zpool create rpool mirror c1t2d0s0 c2t1d0s0
# lucreate -c ufsBE -n zfsBE -p rpool
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <ufsBE>.
Creating initial configuration for primary boot environment <ufsBE>.
The device </dev/dsk/c1t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/c1t0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c1t2d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Populating file systems on boot environment <zfsBE>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Creating compare databases for boot environment <zfsBE>.
Creating compare database for file system </rpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfsBE>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-qD.mnt
updating /.alt.tmp.b-qD.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
```

Após a operação com `lucreate` ter terminado, use o comando `lustatus` para exibir o status do BE. Por exemplo:

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

Em seguida, consulte a lista de componentes do ZFS. Por exemplo:

EXEMPLO 5-3 Utilizando o Oracle Solaris Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS (Continuação)

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                7.17G 59.8G  95.5K  /rpool
rpool/ROOT           4.66G 59.8G   21K  /rpool/ROOT
rpool/ROOT/zfsBE    4.66G 59.8G  4.66G  /
rpool/dump           2G    61.8G  16K   -
rpool/swap           517M  60.3G  16K   -
```

Depois, use o comando `luactivate` para ativar o novo BE do ZFS. Por exemplo:

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

*****

The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.

*****
.
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
```

Depois, reinicialize o sistema para o BE do ZFS.

```
# init 6
```

Comprove que o BE do ZFS está ativo.

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

Se voltar ao BE do UFS, você terá que reimportar os conjuntos de armazenamento ZFS que foram criados enquanto o BE do ZFS estava sendo inicializado, porque tais grupos não estão automaticamente disponíveis no BE do UFS.

Se o BE do UFS já não for mais necessário, você pode removê-lo com o comando `ludelete`.

EXEMPLO 5-4 Utilizando o Oracle Solaris Live Upgrade para criar um BE do ZFS a partir de um BE do ZFS

A criação de um BE do ZFS a partir de um BE do ZFS no mesmo pool é muito rápida porque esta operação usa os recursos de instantâneos e clones do ZFS. Se o BE estiver no mesmo conjunto ZFS, por exemplo, a opção `-p` é omitida.

Se possuir vários BEs de ZFS, faça o seguinte para selecionar a partir de qual BE inicializar:

- SPARC: você pode utilizar o comando `boot -L` para identificar os BEs disponíveis e selecionar um BE o qual deseja inicializar utilizando o comando `boot -Z`.
- x86: você pode selecionar um BE do menu GRUB.

Para obter mais informações, consulte o [Exemplo 5-9](#).

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

EXEMPLO 5-5 Atualizando o BE do ZFS (`luupgrade`)

Você pode atualizar o BE do ZFS com correções ou pacotes adicionais.

O processo básico é:

- Crie um BE alternativo com o comando `lucreate`.
- Ative e inicialize a partir do BE alternativo.
- Atualize o BE principal do ZFS com o comando `luupgrade` para adicionar pacotes ou correções.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                 Complete Now    On Reboot Delete Status
-----
```

EXEMPLO 5-5 Atualizando o BE do ZFS (luupgrade) (Continuação)

```

zfsBE          yes      no      no      yes      -
zfs2BE        yes      yes     yes     no       -
# luupgrade -p -n zfsBE -s /net/system/export/s10up/Solaris_10/Product SUNWchxge
Validating the contents of the media </net/install/export/s10up/Solaris_10/Product>.
Mounting the BE <zfsBE>.
Adding packages to the BE <zfsBE>.

Processing package instance <SUNWchxge> from </net/install/export/s10up/Solaris_10/Product>

Chelsio N110 10GE NIC Driver(sparc) 11.10.0,REV=2006.02.15.20.41
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.

This appears to be an attempt to install the same architecture and
version of a package which is already installed. This installation
will attempt to overwrite this package.

Using </a> as the package base directory.
## Processing package information.
## Processing system information.
   4 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SUNWchxge> [y,n,?] y
Installing Chelsio N110 10GE NIC Driver as <SUNWchxge>

## Installing part 1 of 1.
## Executing postinstall script.

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

```

Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (Solaris 10 10/08)

É possível utilizar o Oracle Solaris Live Upgrade para migrar um sistema com regiões, mas as configurações suportadas são limitadas na versão Solaris 10 10/08. Se estiver instalando ou atualizando para pelo menos o Solaris 10 5/09, mais configurações de região são suportadas. Para mais informações, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(no mínimo Solaris 10 5/09\)”](#) na página 155.

Esta seção descreve como configurar e instalar um sistema com regiões para que possa ser atualizado e ter patches instalados com o Oracle Solaris Live Upgrade. Se for migrar para um

sistema de arquivos raiz ZFS sem regiões, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar para um sistema de arquivos raiz ZFS \(sem regiões\)”](#) na página 145.

Se for migrar ou configurar um sistema com regiões no Solaris 10 10/08, consulte os seguintes procedimentos:

- [“Como migrar um sistema de arquivos raiz UFS com raízes de regiões em UFS para um sistema de arquivos raiz ZFS \(Solaris 10 10/08\)”](#) na página 150
- [“Como configurar um sistema de arquivos raiz ZFS com raízes de região em ZFS \(Solaris 10 10/08\)”](#) na página 152
- [“Como atualizar ou aplicar correções em um sistema de arquivos raiz ZFS com raízes de região em ZFS \(Solaris 10 10/08\)”](#) na página 153
- [“Resolvendo problemas do ponto de montagem ZFS que impedem a inicialização com êxito \(Solaris 10 10/08\)”](#) na página 173

Siga estes procedimentos recomendados para configurar regiões em um sistema com sistemas de arquivos raiz ZFS para garantir que o Live Upgrade possa ser utilizado em tal sistema.

▼ **Como migrar um sistema de arquivos raiz UFS com raízes de regiões em UFS para um sistema de arquivos raiz ZFS (Solaris 10 10/08)**

Este procedimento explica como migrar de um sistema de arquivos raiz UFS com regiões instaladas para um sistema de arquivos de raiz e configuração de raiz de região ZFS que pode ser atualizada ou ter patches instalado.

Nas etapas a seguir, o nome do conjunto de exemplo é `rpool` e o nome de exemplo do ambiente de inicialização ativo é `s10BE*`.

1 Atualize os sistema para o Solaris 10 10/08 se estiver executando uma versão anterior do Solaris 10.

Para informações sobre atualizações de um sistema que executa o Solaris 10, consulte [Oracle Guia de instalação do Solaris 10 9/10: Solaris Live Upgrade e planejamento da atualização](#).

2 Crie o conjunto raiz.

```
# zpool create rpool mirror c0t1d0 c1t1d0
```

Para obter informações sobre os requisitos do conjunto raiz, consulte [“Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS”](#) na página 127.

3 Comprove que as regiões do ambiente do UFS estejam inicializadas.

4 Crie o novo ambiente de inicialização do ZFS.

```
# lucreate -n s10BE2 -p rpool
```

Este comando estabelece os conjuntos de dados no conjunto raiz do novo ambiente de inicialização e copia o ambiente de inicialização atual (incluindo as regiões) em tais conjuntos de dados.

5 Ative o novo ambiente de inicialização do ZFS.

```
# luactivate s10BE2
```

Agora, o sistema está executando um sistema de arquivos raiz ZFS, mas as raízes de região no UFS ainda estão no sistema de arquivos raiz UFS. É necessário completar as próximas etapas para migrar totalmente as regiões UFS para uma configuração ZFS suportada.

6 Reinicialize o sistema.

```
# init 6
```

7 Migre as regiões para um BE do ZFS.

a. Inicialize as regiões.

b. Crie outro BE do ZFS dentro do grupo.

```
# lucreate s10BE3
```

c. Ative o novo ambiente de inicialização.

```
# luactivate s10BE3
```

d. Reinicialize o sistema.

```
# init 6
```

Esta etapa verifica se o BE do ZFS e as regiões estão inicializados.

8 Resolva qualquer potencial problema de ponto de montagem.

Devido a um erro no Oracle Solaris Live Upgrade, o ambiente de inicialização inativo pode falhar ao ser inicializado porque o conjunto de dados ZFS ou o conjunto de dados ZFS da região no ambiente de inicialização possui um ponto de montagem inválido.

a. Revise a saída do `zfs list`.

Procure os pontos de montagem temporários incorretos. Por exemplo:

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

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

O ponto de montagem do BE do ZFS raiz (rpool/ROOT/s10u6) deve ser /.

b. Redefina os pontos de montagem do BE do ZFS e seus conjuntos de dados.

Por exemplo:

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

c. Reinicialize o sistema.

Quando a opção para inicializar um ambiente de inicialização específico for apresentada, tanto no menu GRUB quanto no prompt do OpenBoot PROM, selecione o ambiente de inicialização cujos pontos de montagem acabam de ser corrigidos.

▼ **Como configurar um sistema de arquivos raiz ZFS com raízes de região em ZFS (Solaris 10 10/08)**

Este procedimento explica como configurar um sistema de arquivos raiz ZFS e a configuração da raiz de região ZFS que pode ser atualizada ou ter patches instalados. Nesta configuração, as raízes de região ZFS são criadas como conjuntos de dados ZFS.

Nas etapas a seguir, o nome do conjunto de exemplo é `rpool` e o nome de exemplo do ambiente de inicialização ativo é `s10BE*`. O nome do conjunto de dados das regiões pode ser qualquer nome de conjunto de dados permitido. No exemplo a seguir, o nome do conjunto de dados das regiões é `regiões`.

1 Instale o sistema com uma raiz ZFS utilizando o instalador de texto interativo Solaris ou o método de instalação Solaris JumpStart.

Para informações sobre a instalação de um sistema de arquivos raiz ZFS utilizando o método da instalação inicial ou o método Solaris JumpStart, consulte [“Instalando um sistema de arquivos raiz ZFS \(Instalação inicial\)”](#) na página 130 ou [“Instalando um sistema de arquivos raiz ZFS \(instalação Oracle Solaris JumpStart\)”](#) na página 139.

2 Inicialize o sistema a partir do conjunto raiz recém-criado.

3 Crie um conjunto de dados para agrupar as raízes da região.

Por exemplo:

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones
```

Definir o valor `canmount` para a propriedade `noauto` impede que o conjunto de dados seja montado de forma diferente da ação explícita do Oracle Solaris Live Upgrade e do código de inicialização do sistema.

4 Monte o conjunto de dados de regiões recém-criado.

```
# zfs mount rpool/ROOT/s10BE/zones
```

O conjunto de dados é montado em `/zones`.

5 Crie e monte um conjunto de dados para cada raiz de região.

```
# zfs create -o canmount=noauto rpool/ROOT/s10BE/zones/zonerootA
# zfs mount rpool/ROOT/s10BE/zones/zonerootA
```

6 Defina as permissões apropriadas no diretório raiz da região.

```
# chmod 700 /zones/zonerootA
```

7 Configure a região, definindo o caminho da região da seguinte forma:

```
# zonecfg -z zoneA
zoneA: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zoneA> create
zonecfg:zoneA> set zonepath=/zones/zonerootA
```

Você pode ativar as regiões para que se inicializem automaticamente quando o sistema for inicializado usando a seguinte sintaxe:

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

8 Instale a região.

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

9 Inicialize a região.

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

▼ Como atualizar ou aplicar correções em um sistema de arquivos raiz ZFS com raízes de região em ZFS (Solaris 10 10/08)

Utilize este procedimento quando precisar atualizar ou instalar patches em um sistema de arquivos raiz ZFS com raízes de região em ZFS. Estas atualizações podem ser tanto uma melhora do sistema quanto uma aplicação de patches.

Nas etapas a seguir, newBE é o nome de exemplo do ambiente de inicialização que é atualizado ou tem patches instalados.

1 Crie um ambiente de inicialização para ser atualizado ou aplicar correções.

```
# lucreate -n newBE
```

O ambiente de inicialização existente, incluindo todas as regiões, é clonado. São criados novos conjuntos de dados para cada conjunto de dados do ambiente de inicialização original. Os novos conjuntos de dados são criados no mesmo pool do pool raiz atual.

2 Selecione um dos procedimentos a seguir para atualizar o sistema ou aplicar patches ao novo ambiente de inicialização:

- Atualize o sistema.

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

Onde a opção `-s` especifica a localização de um meio de instalação do Solaris.

- Aplique patches no novo ambiente de inicialização.

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

3 Ative o novo ambiente de inicialização.

```
# luactivate newBE
```

4 Inicialize a partir do ambiente de inicialização recém ativado.

```
# init 6
```

5 Resolva qualquer potencial problema de ponto de montagem.

Devido a um erro no recurso do Oracle Solaris Live Upgrade, o ambiente de inicialização inativo pode falhar ao ser inicializado porque o conjunto de dados ZFS ou o conjunto de dados ZFS da região no ambiente de inicialização possuem um ponto de montagem inválido.

a. Revise a saída do `zfs list`.

Procure os pontos de montagem temporários incorretos. Por exemplo:

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

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

O ponto de montagem do BE do ZFS raiz (`rpool/ROOT/newBE`) deve ser `/`.

b. Redefina os pontos de montagem do BE do ZFS e seus conjuntos de dados.

Por exemplo:

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

c. Reinicialize o sistema.

Quando a opção para inicializar um ambiente de inicialização específico for apresentada, tanto no menu GRUB quanto no prompt do OpenBoot PROM, selecione o ambiente de inicialização cujos pontos de montagem acabam de ser corrigidos.

Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (no mínimo Solaris 10 5/09)

É possível utilizar o recurso Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões a partir da versão Solaris 10 10/08. Configurações adicionais de regiões esparsas ou completas de raiz são suportadas pelo Live Upgrade, começando com a versão Solaris 10 5/09.

Esta seção descreve como configurar e instalar um sistema com regiões para que possa ser atualizado e ter o patch instalado com o Oracle Solaris Live Upgrade a partir da versão Solaris 10 5/09. Se for migrar para um sistema de arquivos raiz ZFS sem regiões, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar para um sistema de arquivos raiz ZFS \(sem regiões\)”](#) na página 145.

Considere os pontos a seguir ao utilizar o Live Upgrade com ZFS e regiões começando com a versão Solaris 10 5/09:

- Para utilizar o Oracle Solaris Live Upgrade com configurações de regiões que são suportadas, começando com, no mínimo, a última versão Solaris 10 5/09, é necessário primeiro atualizar o sistema para a versão Solaris 10 5/09 pelo menos, ao utilizar o programa de atualização padrão.
- Logo, com o Oracle Solaris Live Upgrade, é possível tanto migrar o sistema de arquivos raiz UFS com raízes de região para o sistema de arquivos raiz ZFS quanto atualizar ou instalar patches no sistema de arquivos raiz ZFS e raízes de região.
- Não é possível migrar diretamente configurações de regiões não suportadas de uma versão Solaris 10 anterior à versão Solaris 10 5/09, no mínimo.

Se estiver migrando ou configurando um sistema com regiões começando com a versão Solaris 10 5/09, revise as informações a seguir:

- [“ZFS suportado com informações de configuração de raiz da região \(pelo menos Solaris 10 5/09\)”](#) na página 156
- [“Como criar um BE do ZFS com um sistema de arquivos raiz ZFS e uma raiz de região \(pelo menos Solaris 10 5/09\)”](#) na página 157
- [“Como atualizar ou instalar patch em um sistema de arquivos raiz ZFS com raízes de região \(pelo menos Solaris 10 5/09\)”](#) na página 159
- [“Como migrar um sistema de arquivos raiz UFS com raiz de região para um sistema de arquivos raiz ZFS \(pelo menos Solaris 10 5/09\)”](#) na página 162

ZFS suportado com informações de configuração de raiz da região (pelo menos Solaris 10 5/09)

Revise as configurações de região suportadas antes de utilizar o recurso Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões.

- **Migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS** – São suportadas as seguintes configurações de raízes de região:
 - Em um diretório no sistema de arquivos raiz UFS
 - Em um subdiretório de um ponto de montagem no sistema de arquivos raiz UFS
 - Sistema de arquivos raiz UFS com uma raiz de região em um diretório de sistema de arquivos UFS ou em um subdiretório de um ponto de montagem de sistema de arquivos raiz UFS e um grupo não raiz ZFS com uma raiz de região

A seguinte configuração de região UFS não é suportada: sistema de arquivos raiz UFS que tem uma raiz de região como ponto de montagem.

- **Migrar ou atualizar um sistema de arquivos raiz ZFS** – São suportadas as seguintes configurações de raízes de região:
 - Em um conjunto de dados no pool raiz ZFS. Em alguns casos, se um conjunto de dados não for fornecido à raiz de região de raiz antes da operação do Oracle Solaris Live Upgrade, um conjunto de dados para a raiz de região (zoned) será criado pelo Oracle Solaris Live Upgrade.
 - Em um subdiretório do sistema de arquivos raiz ZFS
 - Em um conjunto de dados fora do sistema de arquivos raiz ZFS
 - Em um subdiretório de um conjunto de dados fora do sistema de arquivos raiz ZFS
 - Em um conjunto de dados em um pool não-raiz. No exemplo a seguir, `zonepool/zones` é um conjunto de dados que contém as raízes da região e `rpool` contém o BE do ZFS:

```
zonepool
zonepool/zones
zonepool/zones/myzone
rpool
rpool/ROOT
rpool/ROOT/myBE
```

A operação do Oracle Solaris Live Upgrade faz instantâneos e clones das regiões em `zonepool` e do BE em `rpool` se for utilizada esta sintaxe:

```
# lucreate -n newBE
```

O BE `newBE` no `rpool/ROOT/newBE` é criado. Quando ativado, `newBE` fornece acesso aos componentes `zonepool`.

No exemplo acima, se `/zonepool/zones` era um subdiretório e não um conjunto de dados separado, então o Live Upgrade iria migrá-las como componentes do conjunto raiz, `rpool`.

- **Migração de regiões ou informações de atualização com regiões para UFS e ZFS:** observe os fatores a seguir que podem afetar tanto a migração quanto a atualização de ambientes UFS e ZFS:
 - Se tiver configurado as regiões conforme descrito em [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões \(Solaris 10 10/08\)” na página 149](#) na versão do Solaris 10 10/08 e atualizado para pelo menos o Solaris 10 5/09, você deve poder migrar para um sistema de arquivos raiz ZFS ou utilizar o Oracle Solaris Live Upgrade a fim de atualizar para no mínimo a versão Solaris 10 5/09.
 - Não crie raízes de regiões em diretórios aninhados, por exemplo `zones/zone1` e `zones/zone1/zone2`. Caso contrário, a montagem poderá falhar no momento de inicialização.

▼ Como criar um BE do ZFS com um sistema de arquivos raiz ZFS e uma raiz de região (pelo menos Solaris 10 5/09)

Utilize este procedimento depois de ter efetuado uma instalação inicial pelo menos da versão Solaris 10 5/09 para criar um sistema de arquivos raiz ZFS. Utilize também este procedimento depois de ter utilizado o recurso `luupgrade` para atualizar um sistema de arquivos raiz ZFS para pelo menos a versão Solaris 10 5/09. Um BE do ZFS criado através deste procedimento pode, portanto, ser atualizado e receber correções.

Nas etapas a seguir, o sistema Oracle Solaris 10 9/10 de exemplo apresenta um sistema de arquivos raiz ZFS e um conjunto de dados de região de raiz em `/rpool/zones`. É criado um BE do ZFS denominado `zfs2BE` que pode ser atualizado e ter patches instalados.

1 Revise os sistemas de arquivos ZFS existentes.

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                7.26G 59.7G  98K    /rpool
rpool/ROOT           4.64G 59.7G  21K    legacy
rpool/ROOT/zfsBE    4.64G 59.7G  4.64G  /
rpool/dump           1.00G 59.7G  1.00G  -
rpool/export         44K   59.7G  23K    /export
rpool/export/home   21K   59.7G  21K    /export/home
rpool/swap           1G    60.7G  16K    -
rpool/zones         633M  59.7G  633M   /rpool/zones
```

2 Certifique-se de que as regiões estejam instaladas e inicializadas.

```
# zoneadm list -cv
ID  NAME      STATUS  PATH                BRAND  IP
0   global    running /                    native shared
2   zfszone   running /rpool/zones        native shared
```

3 Crie o BE do ZFS.

```
# lucreate -n zfs2BE
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <zfsBE>.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <zfsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

4 Ative o BE do ZFS.

```
# lustatus
Boot Environment          Is      Active Active   Can      Copy
Name                     Complete Now    On Reboot Delete Status
-----
zfsBE                    yes     yes   yes     no      -
zfs2BE                   yes     no    no      yes     -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
# init 6
```

5 Comprove se os sistemas de arquivos ZFS e as regiões foram criados no novo BE.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               7.38G 59.6G  98K    /rpool
rpool/ROOT                          4.72G 59.6G  21K    legacy
rpool/ROOT/zfs2BE                   4.72G 59.6G  4.64G  /
rpool/ROOT/zfs2BE@zfs2BE            74.0M  -      4.64G  -
rpool/ROOT/zfsBE                    5.45M 59.6G  4.64G  /.alt.zfsBE
rpool/dump                          1.00G 59.6G  1.00G  -
rpool/export                        44K   59.6G  23K    /export
rpool/export/home                   21K   59.6G  21K    /export/home
rpool/swap                          1G    60.6G  16K    -
rpool/zones                         17.2M 59.6G  633M   /rpool/zones
rpool/zones-zfsBE                   653M  59.6G  633M   /rpool/zones-zfsBE
rpool/zones-zfsBE@zfs2BE            19.9M  -      633M   -
# zoneadm list -cv
ID NAME                STATUS  PATH                                BRAND  IP
```

```

0 global          running /          native shared
- zfszone        installed /rpool/zones native shared

```

▼ Como atualizar ou instalar patch em um sistema de arquivos raiz ZFS com raízes de região (pelo menos Solaris 10 5/09)

Utilize este procedimento quando for necessário atualizar ou instalar patches em um sistema de arquivos raiz ZFS com raízes de região na versão Solaris 10 5/09, no mínimo. Estas atualizações podem ser tanto uma melhora do sistema quanto uma aplicação de patches.

Nas etapas a seguir, `zfs2BE` é o nome de exemplo do ambiente de inicialização no qual se realiza a atualização ou se aplicam as correções.

1 Revise os sistemas de arquivos ZFS existentes.

```

# zfs list
NAME                                USED AVAIL REFER MOUNTPOINT
rpool                                7.38G 59.6G 100K /rpool
rpool/ROOT                           4.72G 59.6G 21K legacy
rpool/ROOT/zfs2BE                     4.72G 59.6G 4.64G /
rpool/ROOT/zfs2BE@zfs2BE              75.0M - 4.64G -
rpool/ROOT/zfsBE                      5.46M 59.6G 4.64G /
rpool/dump                            1.00G 59.6G 1.00G -
rpool/export                          44K 59.6G 23K /export
rpool/export/home                     21K 59.6G 21K /export/home
rpool/swap                             1G 60.6G 16K -
rpool/zones                           22.9M 59.6G 637M /rpool/zones
rpool/zones-zfsBE                     653M 59.6G 633M /rpool/zones-zfsBE
rpool/zones-zfsBE@zfs2BE              20.0M - 633M -

```

2 Certifique-se de que as regiões estejam instaladas e inicializadas.

```

# zoneadm list -cv
ID NAME          STATUS  PATH          BRAND  IP
0 global         running /             native shared
5 zfszone        running /rpool/zones native shared

```

3 Crie um BE do ZFS para ser atualizado ou aplicar as correções.

```

# lucreate -n zfs2BE
Analyzing system configuration.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <zfsBE>.
Creating boot environment <zfs2BE>.
Cloning file systems from boot environment <zfsBE> to create boot environment <zfs2BE>.
Creating snapshot for <rpool/ROOT/zfsBE> on <rpool/ROOT/zfsBE@zfs2BE>.
Creating clone for <rpool/ROOT/zfsBE@zfs2BE> on <rpool/ROOT/zfs2BE>.
Setting canmount=noauto for </> in zone <global> on <rpool/ROOT/zfs2BE>.
Creating snapshot for <rpool/zones> on <rpool/zones@zfs10092BE>.
Creating clone for <rpool/zones@zfs2BE> on <rpool/zones-zfs2BE>.

```

```
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.
```

4 Selecione um dos procedimentos a seguir para atualizar o sistema ou instalar patches ao novo ambiente de inicialização:

- Atualize o sistema.

```
# luupgrade -u -n zfs2BE -s /net/install/export/s10up/latest
```

Onde a opção -s especifica a localização de um meio de instalação do Solaris.

Este processo pode demorar um longo tempo.

Para obter um exemplo completo do processo de luupgrade, consulte o [Exemplo 5-6](#).

- Aplique patches no novo ambiente de inicialização.

```
# luupgrade -t -n zfs2BE -t -s /patchdir patch-id-02 patch-id-04
```

5 Ative o novo ambiente de inicialização.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                 Complete Now    On Reboot Delete Status
-----
zfsBE                 yes     yes   yes     no    -
zfs2BE                yes     no    no      yes   -
# luactivate zfs2BE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfs2BE>.
.
.
.
```

6 Inicialize a partir do ambiente de inicialização recém ativado.

```
# init 6
```

Exemplo 5-6 Atualizando um sistema de arquivos raiz ZFS com raiz de região para o sistema de arquivos raiz ZFS do Oracle Solaris 10 9/10

Neste exemplo, um BE do ZFS (zfsBE), criado em um sistema Solaris 10 10/09 com um sistema de arquivos raiz ZFS em um grupo não-raiz, é atualizado para a versão do Solaris 10 9/10. Este processo pode ser demorado. A seguir, o BE atualizado (zfs2BE) é ativado. Certifique-se de que as regiões estejam instaladas e inicializadas antes de tentar atualizar.

Neste exemplo, o grupo zonepool e o conjunto de dados /zonepool/zones e a região zfszone são criados como segue:

```
# zpool create zonepool mirror c2t1d0 c2t5d0
# zfs create zonepool/zones
# chmod 700 zonepool/zones
# zonecfg -z zfszone
```



```

zfszone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zfszone> create
zonecfg:zfszone> set zonepath=/zonepool/zones
zonecfg:zfszone> verify
zonecfg:zfszone> exit
# zoneadm -z zfszone install
cannot create ZFS dataset zonepool/zones: dataset already exists
Preparing to install zone <zfszone>.
Creating list of files to copy from the global zone.
Copying <8960> files to the zone.
.
.
.

# zoneadm list -cv
ID NAME          STATUS  PATH                               BRAND  IP
  0 global         running /                                   native shared
  2 zfszone        running /zonepool/zones                  native shared

# lucreate -n zfsBE
.
.
.
# luupgrade -u -n zfsBE -s /net/install/export/s10up/latest
40410 blocks
miniroot filesystem is <lofs>
Mounting miniroot at </net/system/export/s10up/latest/Solaris_10/Tools/Boot>
Validating the contents of the media </net/system/export/s10up/latest>.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains <Solaris> version <10>.
Constructing upgrade profile to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE <zfsBE>.
Determining packages to install or upgrade for BE <zfsBE>.
Performing the operating system upgrade of the BE <zfsBE>.
CAUTION: Interrupting this process may leave the boot environment unstable
or unbootable.
Upgrading Solaris: 100% completed
Installation of the packages from this media is complete.
Updating package information on boot environment <zfsBE>.
Package information successfully updated on boot environment <zfsBE>.
Adding operating system patches to the BE <zfsBE>.
The operating system patch installation is complete.
INFORMATION: The file </var/sadm/system/logs/upgrade_log> on boot
environment <zfsBE> contains a log of the upgrade operation.
INFORMATION: The file </var/sadm/system/data/upgrade_cleanup> on boot
environment <zfsBE> contains a log of cleanup operations required.
INFORMATION: Review the files listed above. Remember that all of the files
are located on boot environment <zfsBE>. Before you activate boot
environment <zfsBE>, determine if any additional system maintenance is
required or if additional media of the software distribution must be
installed.
The Solaris upgrade of the boot environment <zfsBE> is complete.
Installing failsafe
Failsafe install is complete.

```

```
# luactivate zfsBE
# init 6
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now   On Reboot Delete Status
-----
zfsBE                  yes     no    no      yes   -
zfs2BE                 yes     yes   yes     no    -
# zoneadm list -cv
ID NAME                STATUS  PATH                                BRAND  IP
0  global               running /                                    native shared
-  zfszone               installed /zonepool/zones                    native  shared
```

▼ Como migrar um sistema de arquivos raiz UFS com raiz de região para um sistema de arquivos raiz ZFS (pelo menos Solaris 10 5/09)

Utilize este procedimento para migrar um sistema com um sistema de arquivos raiz UFS e uma raiz de região para a versão Solaris 10 5/09, pelo menos. Em seguida, utilize o Oracle Solaris Live Upgrade para criar um BE do ZFS.

Nas etapas a seguir, o nome do BE do UFS de exemplo é `c0t1d0s0`, a raiz da região UFS é `zonepool/zfszone`, e o BE de raiz ZFS é `zfs`.

1 Atualize o sistema para, no mínimo, a versão Solaris 10 5/09 se estiver executando uma versão anterior do Solaris 10.

Para mais informações sobre atualizações de um sistema que executa o Solaris 10, consulte [Oracle Guia de instalação do Solaris 10 9/10: Solaris Live Upgrade e planejamento da atualização](#).

2 Crie o conjunto raiz.

Para obter informações sobre os requisitos do conjunto raiz, consulte “[Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS](#)” na página 127.

3 Comprove que as regiões do ambiente do UFS estejam inicializadas.

```
# zoneadm list -cv
ID NAME                STATUS  PATH                                BRAND  IP
0  global               running /                                    native shared
2  zfszone               running /zonepool/zones                    native  shared
```

4 Crie o novo ambiente de inicialização do ZFS.

```
# lucreate -c c1t1d0s0 -n zfsBE -p rpool
```

Este comando estabelece os conjuntos de dados no conjunto raiz do novo ambiente de inicialização e copia o ambiente de inicialização atual (incluindo as regiões) em tais conjuntos de dados.

5 Ative o novo ambiente de inicialização do ZFS.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
c1t1d0s0              yes     no     no      yes   -
zfsBE                 yes     yes    yes     no    -      #
Luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.
.
.
.
```

6 Reinicialize o sistema.

```
# init 6
```

7 Comprove se os sistemas de arquivos ZFS e as regiões foram criados no novo BE.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                6.17G 60.8G   98K    /rpool
rpool/ROOT                           4.67G 60.8G   21K    /rpool/ROOT
rpool/ROOT/zfsBE                     4.67G 60.8G   4.67G  /
rpool/dump                           1.00G 60.8G   1.00G  -
rpool/swap                           517M  61.3G   16K    -
zonepool                             634M  7.62G   24K    /zonepool
zonepool/zones                       270K  7.62G   633M   /zonepool/zones
zonepool/zones-c1t1d0s0              634M  7.62G   633M   /zonepool/zones-c1t1d0s0
zonepool/zones-c1t1d0s0@zfsBE       262K  -        633M   -
# zoneadm list -cv
ID NAME          STATUS  PATH                                BRAND  IP
0  global         running /                                     native shared
-  zfszone        installed /zonepool/zones                    native  shared
```

Exemplo 5-7 Migrando um sistema de arquivos raiz UFS com raiz de região para um sistema de arquivos raiz ZFS

Neste exemplo, um sistema Solaris 10 9/10 com um sistema de arquivos raiz UFS e uma raiz de região (/uzone/ufszzone) e um conjunto não-raiz ZFS (grupo) e a região raiz (/grupo/zfszone) é migrado para um sistema de arquivos raiz ZFS. Certifique-se de que o conjunto raiz ZFS seja criado e que as regiões estejam instaladas e inicializadas antes de realizar a migração.

```
# zoneadm list -cv
ID NAME          STATUS  PATH                                BRAND  IP
0  global         running /                                     native shared
2  ufszone        running /uzone/ufszzone                    native shared
3  zfszone        running /pool/zones/zfszone                native  shared
```

```
# lucreate -c ufsBE -n zfsBE -p rpool
```

```
Analyzing system configuration.
```

```
No name for current boot environment.
```

```

Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/clt0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/clt0d0s0>.
Comparing source boot environment <ufsBE> file systems with the file
file system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/clt1d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <ufsBE>.
Creating boot environment <zfsBE>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
Populating file systems on boot environment <zfsBE>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Copying root of zone <ufszone> to </.alt.tmp.b-EYd.mnt/uzone/ufszone>.
Creating snapshot for <pool/zones/zfszone> on <pool/zones/zfszone@zfsBE>.
Creating clone for <pool/zones/zfszone@zfsBE> on <pool/zones/zfszone-zfsBE>.
Creating compare databases for boot environment <zfsBE>.
Creating compare database for file system </rpool/ROOT>.
Creating compare database for file system </>.
Updating compare databases on boot environment <zfsBE>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-DLd.mnt
updating /.alt.tmp.b-DLd.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.

```

lustatus

Boot Environment Name	Is Complete	Active Now	Active On Reboot	Can Delete	Copy Status
ufsBE	yes	yes	yes	no	-
zfsBE	yes	no	no	yes	-

luactivate zfsBE

```

.
.
.
# init 6
.
.
.

```

zfs list

NAME	USED	AVAIL	REFER	MOUNTPOINT
pool	628M	66.3G	19K	/pool
pool/zones	628M	66.3G	20K	/pool/zones
pool/zones/zfszone	75.5K	66.3G	627M	/pool/zones/zfszone
pool/zones/zfszone-ufsBE	628M	66.3G	627M	/pool/zones/zfszone-ufsBE
pool/zones/zfszone-ufsBE@zfsBE	98K	-	627M	-
rpool	7.76G	59.2G	95K	/rpool
rpool/ROOT	5.25G	59.2G	18K	/rpool/ROOT
rpool/ROOT/zfsBE	5.25G	59.2G	5.25G	/
rpool/dump	2.00G	59.2G	2.00G	-

```

rpool/swap                517M  59.7G   16K   -
# zoneadm list -cv
ID NAME                STATUS   PATH                                BRAND  IP
 0 global              running  /                                    native shared
- ufszone              installed /uzone/ufszone                    native shared
- zfszone              installed /pool/zones/zfszone              native shared

```

Suporte do ZFS para dispositivos de intercâmbio e dump

Durante a instalação inicial do Solaris OS ou depois de efetuar uma migração do Oracle Solaris Live Upgrade de um sistema de arquivos UFS, é criada uma área de permuta em um volume ZFS no conjunto raiz ZFS. Por exemplo:

```

# swap -l
swapfile                dev swaplo blocks  free
/dev/zvol/dsk/rpool/swap 256,1      16 4194288 4194288

```

Durante a instalação inicial do Solaris OS ou do Oracle Solaris Live Upgrade de um sistema de arquivos UFS, é criado um dispositivo de despejo em um volume ZFS no conjunto raiz ZFS. No geral, um dispositivo de despejo não necessita de administração pois é configurado automaticamente no momento da instalação. Por exemplo:

```

# dumpadm
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
Savecore enabled: yes
Save compressed: on

```

Se desativar e remover o dispositivo de despejo, então é necessário ativá-lo com o comando `dumpadm` depois de ser recriado. Na maioria dos casos, será necessário apenas ajustar o tamanho do dispositivo de despejo ao utilizar o comando `zfs`.

Para obter informações sobre os tamanhos dos volumes de permuta e despejo que são criados pelos programas de instalação, consulte [“Requisitos de instalação do Oracle Solaris e Oracle Solaris Live Upgrade para suporte ZFS” na página 127](#).

Os tamanhos do volume de intercâmbio e do volume de dump podem ser ajustados durante e depois da instalação. Para mais informações, consulte [“Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS” na página 166](#).

Considere os problemas a seguir ao trabalhar com dispositivos de permuta e despejo ZFS:

- Volumes ZFS separados devem ser utilizados para área de permuta e os dispositivos de despejo.
- Atualmente, o uso de um arquivo de intercâmbio em um sistema de arquivos ZFS não é suportado.

- Se precisar alterar a área de intercâmbio ou o dispositivo de dump após a instalação ou atualização do sistema, use os comandos `swap` e `dumpadm` como nas versões anteriores do Solaris. Para obter mais informações, consulte o Capítulo 20, “Configuring Additional Swap Space (Tasks),” no *System Administration Guide: Devices and File Systems* e o Capítulo 17, “Managing System Crash Information (Tasks),” no *System Administration Guide: Advanced Administration*.

Consulte as seções a seguir para mais informações:

- “Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS” na página 166
- “Solução de problemas para dispositivo de despejo ZFS” na página 168

Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS

Devido às diferenças no modo em que a instalação da raiz ZFS determina o tamanho dos dispositivos de permuta e despejo, pode ser necessário ajustar o tamanho destes dispositivos antes, durante ou depois da instalação.

- Você pode ajustar o tamanho dos volumes de intercâmbio e dump durante uma instalação inicial. Para obter mais informações, consulte o [Exemplo 5–1](#).
- É possível criar e dimensionar os volumes de permuta e despejo antes de efetuar a operação com o Oracle Solaris Live Upgrade. Por exemplo:

1. Crie o pool de armazenamento.

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

2. Crie o dispositivo de dump.

```
# zfs create -V 2G rpool/dump
```

3. Ative o dispositivo de despejo.

```
# dumpadm -d /dev/zvol/dsk/rpool/dump
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash/t2000
Savecore enabled: yes
Save compressed: on
```

4. Selecione um dos procedimentos seguintes para criar uma área de intercâmbio:

- SPARC: crie sua área de permuta. Defina o tamanho do bloco em 8 KB.

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

- x86: crie a área de permuta. Defina o tamanho do bloco em 4 KB.

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

5. É preciso ativar a área de permuta quando um novo dispositivo de permuta é adicionado ou alterado.
6. Adicione uma entrada para o volume de permuta para o arquivo `/etc/vfstab`.

O Oracle Solaris Live Upgrade não redimensiona os volumes de permuta e despejo existentes.

- Você pode redefinir a propriedade `volsize` do dispositivo de dump após a instalação de um sistema. Por exemplo:

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

- Você pode redimensionar o volume de intercâmbio, mas até que o CR 6765386 esteja integrado, é melhor remover primeiro o dispositivo de intercâmbio. Em seguida, recrie-lo. Por exemplo:

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs volsize=2G rpool/swap
# swap -a /dev/zvol/dsk/rpool/swap
```

Para obter mais informações sobre como remover um dispositivo de permuta em um sistema ativo, consulte este site:

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

- Você pode ajustar o tamanho dos volumes de intercâmbio e dump em um perfil do JumpStart usando a sintaxe de perfil semelhante à seguinte:

```
install_type initial_install
cluster SUNWCXall
pool rpool 16g 2g 2g c0t0d0s0
```

Neste perfil, duas entradas 2g configuram o tamanho do volume de permuta e volume de despejo como 2 GB cada.

- Caso precise de mais espaço de intercâmbio em um sistema que já está instalado, basta adicionar outro volume de intercâmbio. Por exemplo:

```
# zfs create -V 2G rpool/swap2
```

Então, ative o novo volume de intercâmbio. Por exemplo:

```
# 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
```

Finalmente, adicione uma entrada para o segundo volume de permuta para o arquivo `/etc/vfstab`.

Solução de problemas para dispositivo de despejo ZFS

Revise os itens a seguir se tiver problemas em capturar um despejo de memória de sistema ou em redimensionar o dispositivo de despejo.

- Se um despejo de memória não foi criado automaticamente, é possível utilizar o comando `savecore` para salvar o despejo de memória.
- Um volume de despejo é criado automaticamente ao instalar inicialmente um sistema de arquivos raiz ZFS ou migrar para um sistema de arquivos raiz ZFS. Na maioria dos casos, será necessário apenas ajustar o tamanho do volume de despejo se o tamanho do volume padrão for muito pequeno. Por exemplo, em um sistema de memória grande, o tamanho do volume de despejo é aumentado para 40 GB, como a seguir:

```
# zfs set volsize=40G rpool/dump
```

Redimensionando um volume de despejo grande pode ser um processo demorado.

Se, por qualquer razão, é necessário ativar um dispositivo de despejo depois de criar um dispositivo de despejo manualmente, utilize a sintaxe semelhante à seguinte:

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

- Um sistema com memória de 128 GB ou maior precisará de um dispositivo de despejo maior que o dispositivo de despejo que foi criado por padrão. Se o dispositivo de despejo for muito pequeno para capturar um despejo de memória existente, uma mensagem semelhante à seguinte é exibida:

```
# 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
```

Para informações sobre dimensionar os dispositivos de permuta e despejo, consulte [“Planning for Swap Space” no *System Administration Guide: Devices and File Systems*](#).

- Não é possível, atualmente, adicionar um dispositivo de despejo a um grupo com vários dispositivos de primeiro nível. Você verá uma mensagem semelhante a esta:

```
# dumpadm -d /dev/zvol/dsk/datapool/dump
dump is not supported on device '/dev/zvol/dsk/datapool/dump': 'datapool' has multiple top level vdevs
```

Adicione o dispositivo de despejo ao conjunto raiz, que não pode possuir vários dispositivos de primeiro nível.

Iniciando a partir de um sistema de arquivos raiz ZFS

Os sistemas baseados em SPARC e em x86 usam o novo estilo de inicialização com um arquivo de inicialização, que é a imagem de um sistema de arquivos que contém os arquivos necessários para a inicialização. Ao inicializar de um sistema de arquivos raiz ZFS, os nomes de caminho do arquivo de inicialização e do arquivo de kernel são resolvidos no sistema de arquivos raiz selecionado para a inicialização.

Quando um sistema é inicializado para instalação, um disco RAM é utilizado para o sistema de arquivos raiz durante todo o processo de instalação.

A inicialização de um sistema de arquivos ZFS é diferente da inicialização de um sistema de arquivos UFS porque, com o ZFS, o especificador de dispositivos de inicialização identifica um conjunto de armazenamento, e não um único sistema de arquivos raiz. Um pool de armazenamento pode conter vários *conjuntos de dados inicializáveis* ou sistemas de arquivos raiz ZFS. Ao inicializar do ZFS, você deve especificar um dispositivo de inicialização e um sistema de arquivos raiz dentro do pool identificado pelo dispositivo de inicialização.

Por padrão, o conjunto de dados selecionado para inicialização é aquele identificado pela propriedade `boot fs` do conjunto. Esta seleção padrão pode ser ignorada ao especificar um conjunto de dados inicializável alternativo no comando `boot -Z`.

Iniciando a partir de um disco alternativo em um pool raiz ZFS espelhado

Você pode criar um pool raiz ZFS espelhado quando o sistema for instalado ou anexar um disco para criar um pool raiz ZFS espelhado após a instalação. Para obter mais informações, consulte:

- [“Instalando um sistema de arquivos raiz ZFS \(Instalação inicial\)”](#) na página 130
- [“Como criar um conjunto raiz espelhado \(pós-instalação\)”](#) na página 136

Consulte os problemas conhecidos em relação aos pools raiz ZFS espelhados:

- CR 6668666: é necessário instalar as informações de inicialização nos discos anexados adicionais utilizando os comandos `installboot` ou `installgrub` para ativar a inicialização em outros discos no espelho. Se criar um pool raiz ZFS espelhado com o método de instalação inicial, esta etapa não necessária. Por exemplo, se `c0t1d0s0` foi o segundo disco adicionado ao espelho, então os comandos de sintaxe `installboot` ou `installgrub` devem ser:

- SPARC:

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

- x86:

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

- Você pode inicializar de outros dispositivos em um pool raiz espelhado. Dependendo da configuração de hardware, pode ser necessário atualizar o PROM ou o BIOS para especificar outro dispositivo de inicialização.

Por exemplo, é possível inicializar a partir de ambos os discos (c1t0d0s0 ou c1t1d0s0) no conjunto a seguir.

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

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

- SPARC: insira o disco alternativo no prompt ok.

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

Após reinicializar o sistema, comprove o dispositivo de inicialização ativo. Por exemplo:

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

- x86: selecione um disco alternativo no conjunto raiz ZFS espelhado do menu apropriado da BIOS.

Então, utilize sintaxe semelhante à seguinte para confirmar que está sendo inicializado do disco alternativo:

```
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: inicializando a partir de um sistema de arquivos raiz ZFS

Em um sistema com base em SPARC com vários BEs do ZFS, é possível inicializar a partir de qualquer BE utilizando o comando `luactivate`.

Durante o processo de instalação do Solaris OS e do Oracle Solaris Live Upgrade, o sistema de arquivos raiz ZFS é atribuído automaticamente com a propriedade `boot fs`.

Pode haver vários conjuntos de dados inicializáveis dentro de um pool. Por padrão, a entrada do conjunto de dados inicializável no arquivo `/nome-do-pool/boot/menu.lst` é identificada pela propriedade `boot fs` do pool. No entanto, a entrada de menu `.lst` pode conter o comando

`boot fs`, que especifica um conjunto de dados alternativo no pool. Desta forma, o arquivo `menu.lst` pode conter entradas para vários sistemas de arquivos dentro do pool.

Quando um sistema de arquivos é instalado com o sistema de arquivos raiz ZFS ou é migrado para um sistema de arquivos raiz ZFS, uma entrada semelhante à seguinte é adicionada ao arquivo `menu.lst`:

```
title zfsBE
bootfs rpool/ROOT/zfsBE
title zfs2BE
bootfs rpool/ROOT/zfs2BE
```

Quando um novo BE é criado, o arquivo `menu.lst` é atualizado automaticamente.

Em um sistema com base no SPARC, há duas novas opções disponíveis:

- Depois que o BE é ativado, é possível utilizar o comando de inicialização `-L` para exibir uma lista dos conjuntos de dados inicializáveis de um conjunto ZFS. Em seguida, pode selecionar um dos conjuntos de dados inicializáveis da lista. São exibidas as instruções para a inicialização de tal conjunto de dados. É possível inicializar o conjunto de dados selecionado seguindo tais instruções.
- É possível utilizar o comando de inicialização `-Z dataset` para inicializar um conjunto de dados específico ZFS.

EXEMPLO 5-8 SPARC: inicializando a partir de um ambiente de inicialização específico ZFS

Se possuir vários BEs do ZFS em um conjunto de armazenamento ZFS no dispositivo de inicialização do sistema, é possível utilizar o comando `luactivate` para especificar um BE padrão.

Por exemplo, os seguintes BEs do ZFS estão disponíveis conforme descrito pela saída de `lustatus`:

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

Se possui vários BEs do ZFS em seu sistema com base em SPARC, é possível utilizar o comando `boot -L` de um BE que é diferente do BE padrão. Entretanto, um BE que é inicializado de uma sessão `boot -L` não é redefinido como o BE padrão, nem a propriedade `bootfs` atualizada. Se deseja fazer o BE inicializado de uma sessão `boot -L` o BE padrão, então é necessário ativá-lo com o comando `luactivate`.

Por exemplo:

EXEMPLO 5-8 SPARC: inicializando a partir de um ambiente de inicialização específico ZFS
(*Continuação*)

```
ok boot -L
Rebooting with command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0 File and args: -L

1 zfsBE
2 zfs2BE
Select environment to boot: [ 1 - 2 ]: 1
To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfsBE

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

EXEMPLO 5-9 SPARC: inicializando um sistema de arquivos no modo à prova de falhas

Em um sistema com base no SPARC, você pode inicializar a partir do arquivo à prova de falhas localizado em `/platform/`uname -i`/failsafe` da seguinte forma:

```
ok boot -F failsafe
```

Para inicializar um arquivo à prova de falha a partir de um determinado conjunto de dados ZFS, utilize uma sintaxe semelhante à seguinte:

```
ok boot -Z rpool/ROOT/zfsBE -F failsafe
```

x86: inicializando a partir de um sistema de arquivos raiz ZFS

As entradas a seguir são adicionadas ao arquivo `/pool-name/boot/grub/menu.lst` durante o processo de instalação do Solaris OS ou durante a operação do Oracle Solaris Live Upgrade para inicializar automaticamente o ZFS:

```
title Solaris 10 9/10 X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Se o dispositivo identificado pelo GRUB como o dispositivo de inicialização contiver um pool de armazenamento do ZFS, o arquivo `menu.lst` é usado para criar o menu GRUB.

Em um sistema com base no x86 com vários BEs do ZFS, você pode selecionar um BE no menu GRUB. Se o sistema de arquivos raiz correspondente a esta entrada do menu for um conjunto de dados ZFS, a opção seguinte é adicionada:

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

EXEMPLO 5-10 x86: inicializando um sistema de arquivos ZFS

Quando um sistema inicializa de um sistema de arquivos ZFS, o dispositivo raiz é especificado pelo parâmetro de inicialização `-B $ZFS-BOOTFS` nas linhas `kernel` ou `module` na entrada do menu GRUB. Este valor de parâmetro, semelhante a todos os parâmetros especificados pela opção `-B`, é passado ao kernel pelo GRUB. Por exemplo:

```
title Solaris 10 9/10 X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

EXEMPLO 5-11 x86: inicializando um sistema de arquivos no modo à prova de falhas

O arquivo à prova de falhas do x86 é `/boot/x86.miniroot-safe` e pode ser inicializado pela seleção da entrada à prova de falhas do Solaris no menu GRUB. Por exemplo:

```
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

Resolvendo problemas do ponto de montagem ZFS que impedem a inicialização com êxito (Solaris 10 10/08)

A melhor forma de alterar o ambiente de inicialização ativo é usando o comando `luactivate`. Se a inicialização do ambiente ativo falha devido à uma instalação de patch incorreta ou a um erro de configuração, a única forma de inicializar a partir de outro ambiente é selecionando tal ambiente no momento da inicialização. É possível selecionar um BE alternativo no menu GRUB em um sistema com base em x86 ou inicializá-lo explicitamente a partir da PROM em um sistema com base em SPARC.

Devido a um erro no Oracle Solaris Live Upgrade na versão Solaris 10 10/08, o ambiente de inicialização que não estiver ativo pode falhar ao inicializar porque o conjunto de dados ZFS ou

o conjunto de dados ZFS da região no ambiente de inicialização possuem um ponto de montagem inválido. O mesmo erro também impede que o BE seja montado se tiver um conjunto de dados /var diferente.

Se um conjunto de dados de região possui um ponto de montagem inválido, tal ponto de montagem pode ser corrigido realizando as etapas a seguir.

▼ Como resolver problemas de ponto de montagem ZFS

1 Inicialize o sistema a partir de um arquivo à prova de falhas.

2 Importe o pool.

Por exemplo:

```
# zpool import rpool
```

3 Procure os pontos de montagem temporários incorretos.

Por exemplo:

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

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

O ponto de montagem do BE raiz (rpool/ROOT/s10u6) deve ser /.

Se a inicialização estiver falhando devido a problemas com o ponto de montagem /var, procure um ponto de montagem temporário similar do conjunto de dados /var.

4 Redefina os pontos de montagem do BE do ZFS e seus conjuntos de dados.

Por exemplo:

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

5 Reinicialize o sistema.

Quando a opção para inicializar um ambiente de inicialização específico for apresentada, tanto no menu GRUB quanto no prompt do OpenBoot PROM, selecione o ambiente de inicialização cujos pontos de montagem acabam de ser corrigidos.

Iniciando com o propósito de recuperação em um ambiente raiz ZFS

Utilize o procedimento a seguir se necessita inicializar o sistema para que possa recuperar uma senha de raiz perdida ou problema semelhante.

Será necessário inicializar no modo à prova de falhas ou de uma mídia alternativa, dependendo da severidade do erro. No geral, é possível inicializar no modo à prova de falhas para recuperar uma senha de raiz perdida ou desconhecida.

- “Como inicializar o ZFS no modo à prova de falhas” na página 175
- “Como inicializar o ZFS de uma mídia alternativa” na página 176

Se desejar recuperar de um conjunto de raiz ou de uma captura de conjunto de raiz, consulte “Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 176.

▼ Como inicializar o ZFS no modo à prova de falhas

1 Inicialize o modo à prova de falhas.

Em um sistema SPARC:

```
ok boot -F failsafe
```

Em um sistema x86, selecione o modo à prova de falhas no prompt do GRUB.

2 Monte o BE do ZFS no /a quando solicitado:

```
.
.
.
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
Starting shell.
```

3 Altere para o diretório /a/etc.

```
# cd /a/etc
```

4 Se necessário, defina o tipo TERM.

```
# TERM=vt100
# export TERM
```

5 Corrija o arquivo passwd ou shadow.

```
# vi shadow
```

6 Reinicialize o sistema.

```
# init 6
```

▼ Como inicializar o ZFS de uma mídia alternativa

Se um problema impede a inicialização com êxito do sistema ou algum outro problema severo ocorra, será necessário inicializar de um servidor de instalação de rede ou do CD de instalação do Solaris. Importe o conjunto raiz, monte o BE do ZFS e tente resolver o problema.

1 Inicialize de um CD de instalação ou da rede.

- SPARC:

```
ok boot cdrom -s
ok boot net -s
```

Se não utilizar a opção `-s`, será necessário sair do programa de instalação.

- x86: selecione a opção de inicialização de rede ou de inicialização de um CD local.

2 Importe o conjunto raiz e especifique um ponto de montagem alternativo. Por exemplo:

```
# zpool import -R /a rpool
```

3 Monte o BE do ZFS. Por exemplo:

```
# zfs mount rpool/ROOT/zfsBE
```

4 Acesse o conteúdo do BE do ZFS do diretório `/a`.

```
# cd /a
```

5 Reinicialize o sistema.

```
# init 6
```

Recuperando o pool raiz ZFS ou instantâneos do pool raiz

As seções seguintes descrevem como realizar as tarefas abaixo:

- “Como substituir um disco no pool raiz ZFS” na página 176
- “Como criar instantâneos do pool raiz” na página 179
- “Como recriar um pool raiz ZFS e recuperar instantâneos do pool raiz” na página 180
- “Como reverter os instantâneos do pool raiz a partir da inicialização à prova de falhas” na página 182

▼ Como substituir um disco no pool raiz ZFS

Pode ser necessário substituir um disco no pool raiz pelas seguintes razões:

- O conjunto raiz é muito pequeno e deseja substituir o disco menor por um disco maior.

- O disco do conjunto raiz apresenta falhas. Em um grupo não redundante, se o disco estiver falhando de modo que o sistema não inicializa, será necessário realizar a inicialização a partir de uma mídia alternativa, como um CD ou a rede, antes de substituir o disco do conjunto raiz.

Em uma configuração de conjunto raiz espelhado, é possível tentar substituir um disco sem inicializar de uma mídia alternativa. É possível substituir um disco falho ao utilizar o comando `zpool replace`. Ou, se possuir um disco adicional, é possível utilizar o comando `zpool attach`. Consulte o procedimento nesta seção para um exemplo de anexação de um disco adicional e desanexação de um disco de conjunto raiz.

Alguns softwares necessitam que traga um disco off-line e desconfigure-o antes de tentar a operação `zpool replace` para substituir um disco falho. Por exemplo:

```
# zpool offline rpool c1t0d0s0
# cfgadm -c unconfigure c1::dsk/c1t0d0
<Physically remove failed disk c1t0d0>
<Physically insert replacement disk c1t0d0>
# cfgadm -c configure c1::dsk/c1t0d0
# zpool replace rpool c1t0d0s0
# zpool online rpool c1t0d0s0
# zpool status rpool
<Let disk resilver before installing the boot blocks>
SPARC# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t0d0s0
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t9d0s0
```

Em alguns hardwares, não é necessário trazer o disco on-line ou reconfigurá-lo depois de ser inserido.

É necessário identificar os nomes de caminho do dispositivo de inicialização do disco atual e do disco novo para que seja possível testar uma inicialização de teste a partir do disco substituto e também uma inicialização manual a partir do disco existente, caso o disco substituto falhe. Neste exemplo no procedimento a seguir, o nome do caminho para o atual disco de conjunto raiz (`c1t10d0s0`) é:

```
/pci@8,700000/pci@3/scsi@5/sd@a,0
```

O nome do caminho para o disco de inicialização substituto (`c1t9d0s0`) é:

```
/pci@8,700000/pci@3/scsi@5/sd@9,0
```

- 1 Conecte fisicamente o disco de substituição (ou novo).**
- 2 Confirme que o novo disco possui uma legenda SMI e um segmento 0.**

Para obter informações sobre como relegendar um disco reservado para o pool raiz, use o seguinte site:

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

3 Anexe o novo disco ao pool raiz.

Por exemplo:

```
# zpool attach rpool c1t10d0s0 c1t9d0s0
```

4 Confirme os status do pool raiz.

Por exemplo:

```
# zpool status rpool
pool: rpool
state: ONLINE
status: One or more devices is currently being resilvered. The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scrub: resilver in progress, 25.47% done, 0h4m to go
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c1t10d0s0	ONLINE	0	0	0
c1t9d0s0	ONLINE	0	0	0

errors: No known data errors

5 Depois da conclusão do resilvering, aplique os blocos de inicialização ao novo disco.

Utilizando uma sintaxe semelhante à seguinte:

- SPARC:

```
# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t9d0s0
```

- x86:

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t9d0s0
```

6 Verifique se é possível inicializar a partir do novo disco.

Por exemplo, em um sistema com base em SPARC, seria utilizada uma sintaxe semelhante à seguinte:

```
ok boot /pci@8,700000/pci@3/scsi@5/sd@9,0
```

7 Se o sistema se inicializa a partir do novo disco, desanexe o disco antigo.

Por exemplo:

```
# zpool detach rpool c1t10d0s0
```

8 Configure o sistema para que se inicialize automaticamente a partir do disco novo, ou usando o comando eeprom, o comando setenv da PROM de inicialização do SPARC, ou reconfigurando a BIOS do PC.

▼ Como criar instantâneos do pool raiz

É possível criar instantâneos do conjunto raiz com propósito de recuperação. A melhor forma de criar instantâneos de conjunto raiz é efetuar um instantâneo recursivo do conjunto raiz.

O procedimento abaixo cria um instantâneo de conjunto raiz recursivo e armazena o instantâneo como um arquivo em um conjunto em um sistema remoto. No caso de uma falha no conjunto raiz, o conjunto de dados remoto não pode ser montado utilizando o NFS e o arquivo de instantâneo pode ser recebido no conjunto recriado. É possível também armazenar instantâneos de conjunto de raiz como os instantâneos reais em um conjunto em um sistema remoto. Enviar e receber os instantâneos de um sistema remoto é um pouco mais complicado, porque é preciso configurar ssh ou usar rsh enquanto o sistema a ser reparado é inicializado a partir da mini-raiz do SO Solaris.

Para informações sobre o armazenamento remoto e a recuperação de instantâneos de conjunto raiz e para as informações mais recentes sobre a recuperação do conjunto raiz, visite este site:

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

Validando remotamente instantâneos armazenados como arquivos ou instantâneos como um importante passo em uma recuperação de conjunto raiz. Com ambos os métodos, os instantâneos seriam recriados em uma base de rotina, como quando a configuração do conjunto é alterada ou quando o Solaris OS é atualizado.

No procedimento a seguir, o sistema é inicializado a partir do ambiente de inicialização BE.

1 Crie um grupo e sistema de arquivos em um sistema remoto para armazenar os instantâneos.

Por exemplo:

```
remote# zfs create rpool/snaps
```

2 Compartilhe o sistema de arquivos com o sistema local.

Por exemplo:

```
remote# zfs set sharenfs='rw=local-system,root=local-system' rpool/snaps
# share
-@rpool/snaps /rpool/snaps sec=sys,rw=local-system,root=local-system ""
```

3 Crie um instantâneo recursivo do pool raiz.

```
local# zfs snapshot -r rpool@0804
```

```
local# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	6.17G	60.8G	98K	/rpool
rpool@0804	0	-	98K	-
rpool/ROOT	4.67G	60.8G	21K	/rpool/ROOT
rpool/ROOT@0804	0	-	21K	-
rpool/ROOT/zfsBE	4.67G	60.8G	4.67G	/
rpool/ROOT/zfsBE@0804	386K	-	4.67G	-
rpool/dump	1.00G	60.8G	1.00G	-

```
rpool/dump@0804          0      -  1.00G  -  
rpool/swap                517M   61.3G   16K    -  
rpool/swap@0804         0      -   16K    -
```

4 Envie os instantâneos do pool raiz ao sistema remoto.

Por exemplo:

```
local# zfs send -Rv rpool@0804 > /net/remote-system/rpool/snaps/rpool.0804  
sending from @ to rpool@0804  
sending from @ to rpool/swap@0804  
sending from @ to rpool/ROOT@0804  
sending from @ to rpool/ROOT/zfsBE@0804  
sending from @ to rpool/dump@0804
```

▼ Como recriar um pool raiz ZFS e recuperar instantâneos do pool raiz

Neste procedimento, suponha as condições a seguir:

- O conjunto raiz ZFS não pode ser recuperado.
- Os instantâneos do conjunto raiz ZFS estão armazenados em um sistema remoto e são compartilhados no NFS.

Todas as etapas são efetuadas no sistema local.

1 Inicialize a partir de um CD/DVD ou da rede.

- SPARC: selecione um dos seguintes métodos de inicialização:

```
ok boot net -s  
ok boot cdrom -s
```

Se não usar a opção `-s`, você terá que sair do programa de instalação.

- x86: selecione a opção para inicializar a partir do DVD ou da rede. Em seguida, saia do programa de instalação.

2 Monte o conjunto de dados remoto do instantâneo.

Por exemplo:

```
# mount -F nfs remote-system:/rpool/snaps /mnt
```

Se seus serviços de rede não estão configurados, poderá ser necessário especificar o endereço IP do *sistema remoto*.

3 Se o disco do conjunto raiz for substituído e não contiver uma legenda de disco que possa ser utilizada pelo ZFS, será necessário rotular o disco novamente.

Para obter mais informações sobre como rotular um disco novamente, vá ao seguinte site:

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

4 Recrie o pool raiz.

Por exemplo:

```
# zpool create -f -o failmode=continue -R /a -m legacy -o cachefile=
/etc/zfs/zpool.cache rpool c1t1d0s0
```

5 Restaure os instantâneos do pool raiz

Esta etapa pode levar algum tempo. Por exemplo:

```
# cat /mnt/rpool.0804 | zfs receive -Fdu rpool
```

O uso da opção -u significa que o arquivo restaurado não é montado quando a operação do zfs receive terminar.

6 Verifique se os conjuntos de dados do pool raiz foram restaurados.

Por exemplo:

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                6.17G 60.8G   98K    /a/rpool
rpool@0804                            0      -     98K    -
rpool/ROOT                            4.67G 60.8G   21K    /legacy
rpool/ROOT@0804                        0      -     21K    -
rpool/ROOT/zfsBE                       4.67G 60.8G  4.67G  /a
rpool/ROOT/zfsBE@0804                 398K   -     4.67G  -
rpool/dump                             1.00G 60.8G  1.00G  -
rpool/dump@0804                        0      -     1.00G  -
rpool/swap                             517M  61.3G   16K    -
rpool/swap@0804                       0      -     16K    -
```

7 Defina a propriedade bootfs no BE do pool raiz.

Por exemplo:

```
# zpool set bootfs=rpool/ROOT/zfsBE rpool
```

8 Instale os blocos de inicialização no novo disco.

SPARC:

```
# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdisk/c1t1d0s0
x86:
```

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

9 Reinicialize o sistema.

```
# init 6
```

▼ Como reverter os instantâneos do pool raiz a partir da inicialização à prova de falhas

Este procedimento pressupõe que os instantâneos existentes do pool raiz estejam disponíveis. Neste exemplo, eles estão disponíveis no sistema local.

```
# zfs snapshot -r rpool@0804
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                6.17G  60.8G   98K    /rpool
rpool@0804                            0      -     98K    -
rpool/ROOT                            4.67G  60.8G   21K    /rpool/ROOT
rpool/ROOT@0804                        0      -     21K    -
rpool/ROOT/zfsBE                       4.67G  60.8G  4.67G    /
rpool/ROOT/zfsBE@0804                 398K   -   4.67G    -
rpool/dump                             1.00G  60.8G  1.00G    -
rpool/dump@0804                        0      -   1.00G    -
rpool/swap                             517M   61.3G   16K    -
rpool/swap@0804                        0      -     16K    -
```

1 Desligue o sistema e inicie no modo à prova de falhas.

```
ok boot -F failsafe
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
```

Starting shell.

2 Reverta cada instantâneo do conjunto raiz.

```
# zfs rollback rpool@0804
# zfs rollback rpool/ROOT@0804
# zfs rollback rpool/ROOT/zfsBE@0804
```

3 Reinicialize no modo multiusuário.

```
# init 6
```

Gerenciando sistemas de arquivos ZFS do Oracle Solaris

Este capítulo oferece informações detalhadas sobre o gerenciamento de sistemas de arquivos ZFS do Oracle Solaris. Conceitos como layout de sistema de arquivos hierárquico, herança de propriedade, gerenciamento de ponto de montagem automático e interações de compartilhamento estão incluídos neste capítulo.

Este capítulo traz as seguintes seções:

- “Gerenciando de sistemas de arquivos ZFS (visão geral)” na página 183
- “Criando, destruindo e renomeando sistemas de arquivos ZFS” na página 184
- “Introduzindo as propriedades do ZFS” na página 187
- “Consultando informações sobre o sistema de arquivos ZFS” na página 201
- “Gerenciando propriedades do ZFS” na página 203
- “Montando e compartilhando sistemas de arquivos ZFS” na página 208
- “Definindo cotas e reservas do ZFS” na página 215

Gerenciando de sistemas de arquivos ZFS (visão geral)

Um sistema de arquivos ZFS está construído sobre um conjunto de armazenamento. Os sistemas de arquivos podem ser criados e destruídos dinamicamente sem a necessidade de alocar ou formatar espaços em disco subjacentes. Uma vez que os sistemas de arquivos são muito leves, e uma vez que são o ponto central da administração no ZFS, é provável que você crie vários deles.

Os sistemas de arquivos ZFS são administrados com o uso do comando `zfs`. O comando `zfs` oferece um conjunto de subcomandos que realizam operações específicas em sistemas de arquivos. Este capítulo descreve estes subcomandos detalhadamente. Instantâneos, volumes e clones também são gerenciados com esses comandos, porém, neste capítulo, estes recursos são tratados brevemente. Para informações detalhadas sobre instantâneos e clones, consulte [Capítulo 7, “Trabalhando com instantâneos e clones do ZFS do Oracle Solaris”](#). Para informações detalhadas sobre volumes ZFS, consulte [“Volumes ZFS” na página 277](#).

Observação – O termo *conjunto de dados* é usado neste capítulo como um termo genérico para fazer referência a um sistema de arquivos, instantâneo, clone ou volume.

Criando, destruindo e renomeando sistemas de arquivos ZFS

Os sistemas de arquivos ZFS podem ser criados e destruídos com os comandos `zfs create` e `zfs destroy`. Os sistemas de arquivos podem ser renomeados utilizando o comando `zfs rename`.

- [“Criando um sistema de arquivos ZFS” na página 184](#)
- [“Destruindo um sistema de arquivos ZFS” na página 185](#)
- [“Renomeando um sistema de arquivos ZFS” na página 186](#)

Criando um sistema de arquivos ZFS

Os sistemas de arquivos ZFS são criados usando o comando `zfs create`. O subcomando `create` apresenta um único argumento: o nome do sistema de arquivos a ser criado. O nome do sistema de arquivos é especificado como um nome de caminho que começa com o nome do conjunto como se segue:

nome-do-pool/[nome-do-sistema-de-arquivos/]nome-do-sistema-de-arquivos

O nome do conjunto e os nomes iniciais do sistema de arquivos no caminho identificam o local no qual o novo sistema de arquivos será criado na hierarquia. O último nome do caminho identifica o nome do sistema de arquivos que será criado. O nome do sistema de arquivos deve estar de acordo com as convenções de nomeação definidas em [“Requisitos para nomeação de componentes do ZFS” na página 50](#).

No exemplo abaixo, um sistema de arquivos denominado `bonwick` é criado no sistema de arquivos `tank/home`.

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

O ZFS monta automaticamente o sistema de arquivos recém-criado se a criação de tal sistema tiver sido bem-sucedida. Por padrão, os sistemas de arquivos são montados como */conjunto de dados*, usando o caminho fornecido pelo nome do sistema de arquivos no subcomando `create`. Nesse exemplo, o sistema de arquivos `bonwick` recém-criado está montado em `/tank/home/bonwick`. Para mais informações sobre os pontos de montagem gerenciados automaticamente, consulte [“Gerenciando pontos de montagem do ZFS” na página 208](#).

Para mais informações sobre o comando `zfs create`, consulte [`zfs\(1M\)`](#).

Você pode definir as propriedades do sistema de arquivos quando ele é criado.

No exemplo a seguir, um ponto de montagem de `/export/zfs` é criado para o sistema de arquivos `tank/home`:

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

Para mais informações sobre propriedades do sistema de arquivos, consulte [“Introduzindo as propriedades do ZFS” na página 187](#).

Destruindo um sistema de arquivos ZFS

Para destruir um sistema de arquivos ZFS, use o comando `zfs destroy`. O sistema de arquivos destruído é desmontado e descompartilhado automaticamente. Para mais informações sobre montagens ou compartilhamentos gerenciados automaticamente, consulte [“Pontos de montagem automáticos” na página 209](#).

No exemplo a seguir, o sistema de arquivos `tabriz` é destruído:

```
# zfs destroy tank/home/tabriz
```



Cuidado – Nenhum aviso de confirmação é exibido com o subcomando `destroy`. Utilize-o com extrema precaução.

Se o sistema de arquivos que será destruído está ocupado e não pode ser desmontado, o comando `zfs destroy` falha. Para destruir um sistema de arquivos ativo, use a opção `-f`. Utilize essa opção com cuidado já que ela pode desmontar, descompartilhar e destruir sistemas de arquivos ativos, provocando comportamentos inesperados no aplicativo.

```
# zfs destroy tank/home/ahrens
cannot unmount 'tank/home/ahrens': Device busy
```

```
# zfs destroy -f tank/home/ahrens
```

O comando `zfs destroy` também falha se um sistema de arquivos possuir descendentes. Para destruir repetidamente um sistema de arquivos e todos os seus descendentes, use a opção `-r`. Observe que uma destruição recursiva também destrói instantâneos, por isso utilize essa opção com cuidado.

```
# zfs destroy tank/ws
cannot destroy 'tank/ws': filesystem has children
use '-r' to destroy the following datasets:
tank/ws/billm
tank/ws/bonwick
tank/ws/maybee
```

```
# zfs destroy -r tank/ws
```

Se o sistema de arquivos a ser destruído possuir dependentes indiretos, mesmo o comando de destruição recursivo descrito acima falha. Para forçar a destruição de *todos* os dependentes, incluindo os sistemas de arquivos clonados fora da hierarquia, deve ser usada a opção `-R`. Use esta opção com muito cuidado.

```
# zfs destroy -r tank/home/schrock
cannot destroy 'tank/home/schrock': filesystem has dependent clones
use '-R' to destroy the following datasets:
tank/clones/schrock-clone

# zfs destroy -R tank/home/schrock
```



Cuidado – Nenhum aviso de confirmação aparece com as opções `-f`, `-r` ou `-R` para o comando `zfs destroy`, por isso utilize estas opções com precaução.

Para mais informações sobre instantâneos e clones, consulte [Capítulo 7, “Trabalhando com instantâneos e clones do ZFS do Oracle Solaris”](#).

Renomeando um sistema de arquivos ZFS

Os sistemas de arquivos podem ser renomeados com o uso do comando `zfs rename`. Com o subcomando `rename` é possível efetuar as operações a seguir:

- Alterar o nome de um sistema de arquivos.
- Realocar o sistema de arquivos dentro da hierarquia ZFS.
- Alterar o nome de um sistema de arquivos e realocá-lo com a hierarquia ZFS.

O exemplo a seguir utiliza o subcomando `rename` para renomear um sistema de arquivos de `kustarz` para `kustarz_old`:

```
# zfs rename tank/home/kustarz tank/home/kustarz_old
```

O exemplo a seguir mostra como utilizar o `zfs rename` para realocar um sistema de arquivos:

```
# zfs rename tank/home/maybee tank/ws/maybee
```

Neste exemplo, o sistema de arquivos `maybee` é realocado de `tank/home` para `tank/ws`. Quando um sistema de arquivos é realocado por meio de renomeação, o novo local deve estar dentro do mesmo conjunto e possuir espaço em disco suficiente para conter esse novo sistema de arquivos. Se o novo local não possuir espaço em disco suficiente, possivelmente por ter ultrapassado sua cota, a operação `rename` falha.

Para mais informações sobre as cotas, consulte [“Definindo cotas e reservas do ZFS” na página 215](#).

A operação `rename` tenta uma sequência de desmontagem/remontagem no sistema de arquivos e nos sistemas de arquivos descendentes. O comando `rename` falha se a operação não puder desmontar um sistema de arquivos ativo. Se ocorre um problema, desmonte o sistema de arquivos a força.

Para obter informações sobre a renomeação de instantâneos, consulte [“Renomeando instantâneos do ZFS” na página 224](#).

Introduzindo as propriedades do ZFS

As propriedades são o mecanismo principal usado para controlar o comportamento de sistemas de arquivos, volumes, instantâneos e clones. A menos que fique estabelecido de outra forma, as propriedades definidas na seção são válidas para todos os tipos de conjunto de dados.

- [“Propriedades nativas somente leitura do ZFS” na página 195](#)
- [“Propriedades nativas definíveis do ZFS” na página 197](#)
- [“Propriedades de usuário do ZFS” na página 200](#)

As propriedades estão divididas em dois tipos: propriedades nativas e propriedades definidas pelo usuário. As propriedades nativas exportam estatísticas internas ou controlam o sistema de arquivos ZFS. Além disso, as propriedades nativas ou são definíveis ou somente leitura. As propriedades de usuário não afetam o comportamento do ZFS, mas podem ser usadas para anotar conjuntos de dados de forma significativa no ambiente. Para mais informações sobre as propriedades de usuário, consulte [“Propriedades de usuário do ZFS” na página 200](#).

A maioria das propriedades configuráveis também são herdáveis. Uma propriedade herdável é uma propriedade que, quando configurada no conjunto de dados pai, é propagada para todos os seus descendentes.

Todas as propriedades herdáveis possuem uma origem associada que indica como a propriedade foi obtida. A origem de uma propriedade pode ter os seguintes valores:

`local`

Indique que a propriedade foi definida explicitamente no conjunto de dados utilizando o comando `zfs set` conforme descrito em [“Definindo propriedades do ZFS” na página 203](#).

`inherited from nome-do-conjunto-de-dados`

Indique que a propriedade foi herdada do predecessor nomeado.

`default`

Indique que o valor de propriedade não foi herdado ou configurado localmente. Essa origem é resultado de nenhum predecessor possuir a propriedade definida como origem `local`.

A tabela seguinte identifica as propriedades nativas somente leitura e definíveis dos sistema de arquivos ZFS. As propriedades nativas somente leitura são identificadas como `tal`. Todas as outras propriedades nativas listadas nesta tabela são definíveis. Para obter informações sobre as propriedades de usuário, consulte [“Propriedades de usuário do ZFS” na página 200](#).

TABELA 6-1 Descrições de propriedades nativas do ZFS

Nome da propriedade	Tipo	Valor padrão	Descrição
<code>aclinherit</code>	Sequência	<code>secure</code>	Controla como as entradas ACL são herdadas quando os arquivos e diretórios são criados. Os valores são <code>discard</code> , <code>noallow</code> , <code>secure</code> e <code>passthrough</code> . Para uma descrição desses valores, consulte “Propriedades da ACL” na página 242 .
<code>aclmode</code>	Sequência	<code>groupmask</code>	Controla como uma entrada ACL é modificada durante uma operação <code>chmod</code> . Os valores são <code>discard</code> , <code>groupmask</code> e <code>passthrough</code> . Para uma descrição desses valores, consulte “Propriedades da ACL” na página 242 .
<code>atime</code>	Booleano	<code>on</code>	Controla se o tempo de acesso dos arquivos é atualizado quando eles são lidos. A desativação dessa propriedade evita o tráfego de gravação de produção durante a leitura de arquivos e pode proporcionar melhoras significativas no desempenho, embora possa confundir utilitários de correio e similares.
<code>available</code>	Número	N/D	<p>A propriedade de somente leitura que identifica a quantidade de espaço em disco disponível no conjunto de dados e todos os seus filhos, mas não assume nenhuma outra atividade no conjunto. Como o espaço em disco é compartilhado dentro de um conjunto, o espaço disponível pode ser limitado por vários fatores, inclusive tamanho físico do conjunto, cotas, reservas ou outros conjuntos de dados dentro do conjunto.</p> <p>A abreviatura da propriedade é <code>avail</code>.</p> <p>Para mais informações sobre contabilidade de espaço em disco, consulte “Contabilidade de espaço em disco do ZFS” na página 62.</p>

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
canmount	Booleano	on	<p>Controla se um sistema de arquivos pode ser montado com o comando <code>zfs mount</code>. Essa propriedade pode ser definida em qualquer sistema de arquivos e a propriedade em si não pode ser herdada. No entanto, quando essa propriedade estiver definida como <code>off</code>, um ponto de montagem pode ser herdado pelos sistemas de arquivos descendentes, mas o sistema de arquivos em si nunca é montado.</p> <p>Quando a opção <code>noauto</code> estiver definida, um conjunto de dados pode ser montado e desmontado apenas explicitamente. O conjunto de dados não é montado automaticamente ao ser criado ou importado, e também não é montado pelo comando <code>zfs mount -a</code> nem desmontado pelo comando <code>zfs unmount -a</code>.</p> <p>Para mais informações, consulte “A propriedade canmount” na página 198.</p>
soma de verificação	Sequência	on	<p>Controla a soma de verificação usada para verificar a integridade dos dados. O valor padrão é <code>on</code>, que seleciona automaticamente um algoritmo apropriado, atualmente <code>fletcher4</code>. Os valores são <code>on</code>, <code>off</code>, <code>fletcher2</code>, <code>fletcher4</code> e <code>sha256</code>. O valor de <code>off</code> desativa a verificação da integridade dos dados do usuário. Não se recomenda o valor <code>off</code>.</p>
compression	Sequência	off	<p>Ativa ou desativa a compactação de um conjunto de dados. Os valores são <code>on</code>, <code>off</code>, <code>lzjb</code>, <code>gzip</code> e <code>gzip-N</code>. Nesse momento, definir esta propriedade como <code>lzjb</code>, <code>gzip</code> ou <code>gzip-N</code> tem o mesmo efeito que defini-la como <code>on</code>. A ativação da compactação em um sistema de arquivos com dados existentes compacta somente os novos dados. Os dados existentes permanecem descompactados.</p> <p>A abreviatura da propriedade é <code>compress</code>.</p>
compressratio	Número	N/D	<p>Propriedade de somente leitura que identifica a taxa de compactação atingida para um conjunto de dados, expressa como um multiplicador. A compactação pode ser ativada pelo <code>zfs set compression=on conjunto de dados</code> comando.</p> <p>O valor é calculado a partir do tamanho lógico de todos os arquivos e da quantidade de dados físicos referenciados. Isso inclui economias explícitas através do uso da propriedade <code>compression</code>.</p>

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
<code>copies</code>	Número	1	Define o número de cópias de dados do usuário por sistema de arquivos. Os valores disponíveis são 1, 2 ou 3. Essas cópias são adicionais a qualquer redundância de nível de pool. O espaço em disco utilizado por várias cópias de dados do usuário é cobrado até o arquivo, conjunto de dados e contagens correspondentes contra cotas e reservas. Além disso, a propriedade <code>used</code> é atualizada quando várias cópias são ativadas. Leve em consideração definir essa propriedade quando o sistema de arquivos for criado porque alterá-la em um sistema de arquivos existente afeta somente os dados recém-gravados.
<code>creation</code>	Seqüência	N/D	A propriedade de somente leitura identifica a data e a hora em que um conjunto de dados foi criado.
<code>devices</code>	Booleano	<code>on</code>	Controle se os arquivos do dispositivo no sistema de arquivos podem ser abertos.
<code>exec</code>	Booleano	<code>on</code>	Controle se os programas dentro de um sistema de arquivos possuem permissão para serem executados. Também, quando definido como <code>off</code> , as chamadas de <code>mmap(2)</code> com <code>PROT_EXEC</code> não são permitidas.
<code>mounted</code>	Booleano	N/D	Propriedade de somente leitura que indica se um sistema de arquivos, clone ou instantâneo estão atualmente montados. Esta propriedade não se aplica aos volumes. O valor pode ser <code>sim</code> ou <code>não</code> .
<code>mountpoint</code>	Seqüência	N/D	Controla o ponto de montagem usado neste sistema de arquivos. Quando a propriedade <code>mountpoint</code> é alterada para um sistema de arquivos, o sistema de arquivos e os descendentes que herdaram o ponto de montagem são desmontados. Se o novo valor for <code>legacy</code> , então eles permanecem desmontados. Do contrário, serão remontados automaticamente no novo local se a propriedade anterior era <code>legacy</code> ou <code>none</code> ou se foram montados antes de a propriedade ter sido alterada. Além disso, quaisquer sistemas de arquivos compartilhados serão descompartilhados e compartilhados no novo local. Para obter mais informações sobre o uso desta propriedade, consulte “Gerenciando pontos de montagem do ZFS” na página 208.

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
<code>primarycache</code>	Sequência	<code>tudo</code>	<p>Controle o que é armazenado no cache primário (ARC). Os valores possíveis são <code>all</code>, <code>none</code> e <code>metadata</code>. Se definido como <code>all</code>, os dados do usuário e os metadados serão colocado no cache. Se definir como <code>nenhum</code>, nem os dados do usuário, nem os metadados serão armazenados em cache. Se configurar para <code>metadados</code>, somente os metadados serão armazenados.</p>
<code>origin</code>	Sequência	<code>N/D</code>	<p>Propriedade somente leitura para volumes ou sistemas de arquivos clonados que identifica o instantâneo a partir do qual o clone foi criado. A origem não pode ser destruída (mesmo com as opções <code>-r</code> ou <code>-f</code>) enquanto o clone existir.</p> <p>Os sistemas de arquivos não clonados apresentam uma origem de <code>nenhum</code>.</p>
<code>quota</code>	Número (ou <code>none</code>)	<code>none</code>	<p>Limita a quantidade de espaço em disco que um conjunto de dados e seus descendentes podem utilizar. Essa propriedade impõe um limite de disco rígido na quantidade do espaço em disco utilizado, incluindo todo o espaço utilizado pelos descendentes, inclusive os sistemas de arquivos e os instantâneos. A definição de uma cota em um descendente de um sistema de arquivos que já possui uma cota, não substitui a cota do antepassado, mas sim impõe um limite adicional. As cotas não podem ser definidas em volumes, já que a propriedade <code>volsize</code> age como uma cota implícita.</p> <p>Para mais informações sobre a definição de cotas, consulte “Definindo cotas em sistemas de arquivos ZFS” na página 216.</p>
<code>readonly</code>	Booleano	<code>off</code>	<p>Controla se um conjunto de dados pode ser modificado. Ao definir como <code>on</code>, nenhuma modificação pode ser feita.</p> <p>A abreviatura da propriedade é <code>rdonly</code>.</p>
<code>recordsize</code>	Número	<code>128K</code>	<p>Especifica um tamanho do bloco sugerido para os arquivos no sistema de arquivos.</p> <p>A abreviatura da propriedade é <code>rcsize</code>. Para uma descrição detalhada, consulte “A propriedade <code>recordsize</code>” na página 199.</p>

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
referenced	Número	N/D	<p>Propriedade de somente leitura que identifica a quantidade de dados acessíveis por um conjunto de dados que pode, ou não, ser compartilhado com outros conjuntos de dados no conjunto.</p> <p>Quando um instantâneo ou um clone é criado, inicialmente faz referência a mesma quantidade de espaço em disco do sistema de arquivos ou do instantâneo a partir do qual foi criado, pois seus conteúdos são idênticos.</p> <p>A abreviatura da propriedade é <i>refer</i>.</p>
refquota	Número (ou none)	none	<p>Define a quantidade de espaço em disco que um conjunto de dados pode consumir. Essa propriedade reforça um limite rígido na quantidade de espaço usado. Esse limite de disco rígido não inclui o espaço em disco utilizado pelos descendentes, como instantâneos e clones.</p>
refreservation	Número (ou none)	none	<p>Define a quantidade mínima de espaço em disco que é garantido para um conjunto de dados, não incluindo seus descendentes, como instantâneos e clones.</p> <p>Quando a quantidade de espaço em disco utilizada está abaixo desse valor, o conjunto de dados é tratado como se estivesse ocupando a quantidade de espaço especificada por <i>refreservation</i>. As reservas de <i>refreservation</i> são calculadas no espaço em disco utilizado do conjunto de dados pai e contadas em relação às cotas e reservas do conjunto de dados pai.</p> <p>Se <i>refreservation</i> estiver configurado, um instantâneo é permitido somente se houver disponível espaço livre suficiente no conjunto fora da reserva para alojar o número atual de bytes <i>referenciados</i> no conjunto de dados.</p> <p>A abreviatura da propriedade é <i>refreserv</i>.</p>

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
<code>reservation</code>	Número (ou none)	<code>none</code>	<p>Configure a quantidade mínima de espaço em disco garantida para um conjunto de dados e seus descendentes. Quando a quantidade de espaço em disco utilizada está abaixo desse valor, o conjunto de dados é tratado como se estivesse utilizando a quantidade de espaço especificada por sua reserva. As reservas são calculadas no espaço em disco do conjunto de dados pai utilizado e contadas em relação às cotas e reservas do conjunto de dados pai.</p> <p>A abreviatura da propriedade é <code>reserv</code>.</p> <p>Para mais informações, consulte “Definindo reservas nos sistemas de arquivos ZFS” na página 219.</p>
<code>secondarycache</code>	Sequência	<code>tudo</code>	<p>Controla o que é armazenado no cache secundário (L2ARC). Os valores possíveis são <code>all</code>, <code>none</code> e <code>metadata</code>. Se definido como <code>all</code>, os dados do usuário e os metadados serão colocado no cache. Se definir como <code>nenhum</code>, nem os dados do usuário, nem os metadados serão armazenados em cache. Se configurar para <code>metadata</code>, somente metadados são armazenados em cache.</p>
<code>setuid</code>	Booleano	<code>on</code>	<p>Controla se o bit <code>setuid</code> é respeitado nos sistemas de arquivos.</p>
<code>shareiscsi</code>	Sequência	<code>off</code>	<p>Controla se um volume ZFS está compartilhado como um destino iSCSI. Os valores das propriedades são <code>on</code>, <code>off</code> e <code>type=disk</code>. Se você desejar configurar o <code>shareiscsi=on</code> para um sistema de arquivos, então todos os volumes ZFS dentro do sistema de arquivos serão compartilhados por padrão. No entanto, ao configurar essas propriedades em um sistema de arquivos não possuirá nenhum efeito direto.</p>
<code>sharenfs</code>	Sequência	<code>off</code>	<p>Controla se o sistema de arquivos está disponível no NFS e quais opções são utilizadas. Se for definido como <code>on</code>, o comando <code>zfs share</code> é chamado sem opções. Do contrário, o comando <code>zfs share</code> é chamado com opções equivalentes ao conteúdo desta propriedade. Se for definida como <code>off</code>, o sistema de arquivos é gerenciado com os comandos <code>share</code> e <code>unshare</code> de legado e com o arquivo <code>dfstab</code>.</p> <p>Para mais informações sobre compartilhamento de sistemas de arquivos ZFS, consulte “Compartilhando e descompartilhando sistemas de arquivos ZFS” na página 213.</p>

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
snapdir	Seqüência	hidden	Controla se o diretório <code>.zfs</code> está oculto ou visível na raiz do sistema de arquivos. Para mais informações sobre o uso de instantâneos, consulte “Visão geral dos instantâneos do ZFS” na página 221 .
type	Seqüência	N/D	Propriedade somente leitura que identifica o tipo de conjunto de dados como <code>filesystem</code> (sistema de arquivos ou clone), <code>volume</code> ou <code>snapshot</code> .
used	Número	N/D	Propriedade de somente leitura que identifica a quantidade de espaço em disco utilizada pelo conjunto de dados e todos os seus descendentes. Para uma descrição detalhada, consulte “A propriedade used” na página 196 .
usedbychildren	Número	off	A propriedade de somente leitura identifica a quantidade de espaço em disco utilizada pelos filhos desse conjunto de dados, que seria liberado se os filhos do conjunto de dados fossem destruídos. A abreviatura da propriedade é <code>usedchild</code> .
usedbydataset	Número	off	A propriedade de somente leitura identifica a quantidade de espaço em disco utilizada por esse próprio conjunto de dados, que seria liberado se o conjunto de dados fosse destruído, depois de destruir quaisquer instantâneos e remover quaisquer reservas <code>reservation</code> . A abreviatura da propriedade é <code>usedds</code> .
usedbyreservation	Número	off	A propriedade de somente leitura identifica a quantidade de espaço em disco que é utilizada por uma <code>reservation</code> definida em um conjunto de dados, que seria liberado se a <code>reservation</code> fosse removida. A abreviatura da propriedade é <code>usedreserv</code> .
usedbysnapshots	Número	off	A propriedade de somente leitura identifica a quantidade de espaço em disco que é utilizado por instantâneos de um conjunto de dados. Em particular, essa é a quantidade de espaço em disco que seria liberado se todos os instantâneos dos conjunto de dados fossem destruídos. Observe que esse valor não é simplesmente a soma das propriedades utilizadas dos instantâneos, porque o espaço pode ser compartilhado por instantâneos múltiplos. A abreviatura da propriedade é <code>usedsnap</code> .

TABELA 6-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da propriedade	Tipo	Valor padrão	Descrição
<code>version</code>	Número	N/D	Identifique a versão em disco do sistema de arquivos, que é independente da versão do conjunto. Essa propriedade pode somente ser configurada para uma versão posterior que esteja disponível a partir da versão do software suportado. Para mais informação, consulte o comando <code>zfs upgrade</code> .
<code>volsize</code>	Número	N/D	Para volumes, especifica o tamanho lógico do volume. Para uma descrição mais detalhada, consulte “ A propriedade <code>volsize</code> ” na página 199.
<code>volblocksize</code>	Número	8 KB	Para volumes, especifica o tamanho do bloco do volume. O tamanho do bloco não pode ser alterado depois que o volume foi gravado, portanto, defina o tamanho do bloco no momento da criação do volume. O tamanho padrão do bloco para volumes é de 8 KB. Qualquer potência de 2 a partir de 512 bytes até 128 KB é válida. A abreviatura da propriedade é <code>volblock</code> .
<code>zoned</code>	Booleano	N/D	Indica se um conjunto de dados foi adicionado à região não global. Se esta propriedade estiver definida, o ponto de montagem não é respeitado na região global e o ZFS não pode montar tal sistema de arquivos quando solicitado. Quando uma região é instalada pela primeira vez, esta propriedade é definida para todos os sistemas de arquivos adicionados. Para obter mais informações sobre o uso do ZFS com regiões instaladas, consulte “ Usando ZFS em um sistema Solaris com regiões instaladas ” na página 280.
<code>xattr</code>	Booleano	on	Indica se os atributos estendidos estão ativados (on) ou desativados (off) para esse sistema de arquivos.

Propriedades nativas somente leitura do ZFS

As propriedades nativas de somente leitura podem ser recuperadas mas não configuradas. As propriedades nativas somente leitura não são herdadas. Algumas propriedades nativas são específicas para um determinado tipo de conjunto de dados. Nesses casos, um tipo de conjunto de dados é mencionado na descrição em [Tabela 6-1](#).

As propriedades nativas de somente leitura estão listadas aqui e são descritas em [Tabela 6-1](#).

- `available`
- `compressratio`

- creation
- mounted
- origin
- referenced
- type
- used

Para obter informações detalhadas, consulte [“A propriedade used” na página 196](#).

- usedbychildren
- usedbydataset
- usedbyreservation
- usedbysnapshots

Para mais informações sobre a contabilidade de espaço em disco, incluindo as propriedades utilizadas, referenciadas e disponíveis, consulte [“Contabilidade de espaço em disco do ZFS” na página 62](#).

A propriedade used

A propriedade utilizada é uma propriedade de somente leitura que identifica a quantidade de espaço em disco utilizada pelo conjunto de dados e todos seus descendentes. Este valor é verificado de acordo com a cota e a reserva do conjunto de dados. O espaço em disco utilizado não inclui a reserva do conjunto de dados, mas leva em consideração a reserva dos conjuntos de dados descendentes. A quantidade de espaço em disco que um conjunto de dados utiliza a partir do pai, assim como a quantidade de espaço em disco que é liberada quando o conjunto de dados é destruído recursivamente, é a maior de seu espaço utilizado e de sua reserva.

Quando os instantâneos são criados, seu espaço em disco é inicialmente compartilhado entre o instantâneo e o sistema de arquivos e possivelmente com os instantâneos anteriores. Conforme o sistema de arquivos é alterado, o espaço previamente compartilhado torna-se único para o instantâneo e é incluído no espaço do instantâneo utilizado. O espaço em disco utilizado por um instantâneo representa seus dados únicos. Adicionalmente, a exclusão de instantâneos pode aumentar a quantidade de espaço em disco único para (e ser utilizada por) outros instantâneos. Para obter mais informações sobre instantâneos e problemas de espaço, consulte [“Comportamento por espaço excedido” na página 62](#).

A quantidade do espaço em disco utilizado, disponível e referenciado não inclui alterações pendentes. As alterações pendentes são consideradas em geral depois de alguns segundos. A realização de uma alteração no disco utilizando a função `fsync(3c)` ou `O_SYNC` não garante necessariamente que as informações sobre o uso de espaço em disco sejam atualizadas imediatamente.

As informações das propriedades `usedbychildren`, `usedbydataset`, `usedbyreservation` e `usedbysnapshots` podem ser exibidas com o comando `zfs list -o space`. Essas propriedades identificam a propriedade utilizada no espaço em disco que é consumido por descendentes. Para obter mais informações, consulte a [Tabela 6-1](#).

Propriedades nativas definíveis do ZFS

As propriedades nativas definíveis são propriedades cujos valores podem ser recuperados e alterados. As propriedades nativas definíveis são definidas usando o comando `zfs set`, conforme descrito em [“Definindo propriedades do ZFS” na página 203](#) ou usando o comando `zfs create` conforme descrito em [“Criando um sistema de arquivos ZFS” na página 184](#). Com exceção das cotas e reservas, as propriedades nativas definíveis são herdadas. Para obter mais informações sobre cotas e reservas, consulte [“Definindo cotas e reservas do ZFS” na página 215](#).

Algumas propriedades nativas definíveis são específicas para um determinado tipo de conjunto de dados. Nesses casos, um tipo de conjunto de dados é mencionado na descrição em [Tabela 6-1](#). Se não for especificamente mencionado, uma propriedade se aplica a todos os tipos de conjunto de dados: sistemas de arquivos, volumes, clones e instantâneos.

As propriedades configuráveis estão listadas aqui e são descritas em [Tabela 6-1](#).

- `aclinherit`
Para uma descrição mais detalhada, consulte [“Propriedades da ACL” na página 242](#).
- `aclmode`
Para uma descrição mais detalhada, consulte [“Propriedades da ACL” na página 242](#).
- `atime`
- `canmount`
- soma de verificação
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`
Para uma descrição mais detalhada, consulte [“A propriedade recordsize” na página 199](#).
- `refquota`

- `refreservation`
- `reservation`
- `secondarycache`
- `shareiscsi`
- `sharenfs`
- `setuid`
- `snapdir`
- `version`
- `volsize`

Para uma descrição mais detalhada, consulte [“A propriedade `volsize`” na página 199](#).

- `volblocksize`
- `zoned`
- `xattr`

A propriedade `canmount`

Se a propriedade `canmount` for definida como `off`, o sistema de arquivos não pode ser montado utilizando os comandos `zfs mount` ou `zfs mount -a`. Configurar essa propriedade para `off` é semelhante a configurar a propriedade `mountpoint` como `nenhum`, exceto que o conjunto de dados ainda apresenta uma propriedade `mountpoint` normal que pode ser herdada. Por exemplo, é possível definir essa propriedade como `off`, estabelecer propriedades herdáveis para os sistemas de arquivos descendentes, mas o sistema de arquivos pai em si nunca é montado nem pode ser acessado pelos usuários. Nesse caso, o sistema de arquivos pai serve como um *recipiente*, de modo que é possível definir propriedades no contêiner, mas o contêiner em si nunca está acessível.

No exemplo a seguir, `userpool` é criado e sua propriedade `canmount` é definida como `desativadaoff`. Os pontos de montagem dos sistemas de arquivos descendentes de usuário são definidos como um ponto de montagem comum, `/export/home`. As propriedades que estão definidas no sistema de arquivos pai são herdadas pelos sistemas de arquivos descendentes, mas o sistema de arquivos pai em si nunca é montado.

```
# 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
```

Se a propriedade `canmount` for definida como `noauto`, significa que o conjunto de dados pode ser montado apenas explicitamente, não automaticamente. Essa configuração é utilizada pelo software de atualização do Oracle Solaris de modo que somente os conjuntos de dados pertencentes ao ambiente de inicialização ativo são montados no momento da inicialização.

A propriedade `recordsize`

A propriedade `recordsize` especifica um tamanho de bloco sugerido para os arquivos no sistema de arquivos.

Esta propriedade é designada unicamente para uso com volumes de trabalho de banco de dados que acessam arquivos em registros de tamanho fixo. O ZFS ajusta automaticamente os tamanhos do bloco de acordo com os algoritmos internos otimizados para padrões de acesso típicos. Para bancos de dados que criam arquivos muito grandes, mas que acessam os arquivos em pequenos blocos aleatórios, esses algoritmos podem ser ideais. Especificar um valor `recordsize` maior ou igual ao tamanho do registro do banco de dados pode resultar em melhoras significativas no desempenho. A utilização dessa propriedade para sistemas de arquivos com fins gerais é totalmente desaconselhado e pode afetar negativamente o desempenho. O tamanho especificado deve ser uma potência de 2 maior ou igual a 512 bytes e menor ou igual a 128 KB. Alterar o valor `recordsize` do sistema de arquivos afeta somente os arquivos criados posteriormente. Os arquivos existentes não são afetados.

A abreviatura da propriedade é `rcsize`.

A propriedade `volsize`

A propriedade `volsize` especifica o tamanho lógico do volume. Por padrão, a criação de um volume estabelece uma reserva para a mesma quantidade. Quaisquer alterações do `volsize` se refletem em uma alteração equivalente na reserva. Estas verificações são usadas para evitar comportamentos inesperados para os usuários. Um volume que contém menos espaço do que ele exige pode resultar em um comportamento indefinido ou corrupção de dados, dependendo de como o volume é usado. Esses efeitos também podem ocorrer quando o tamanho do volume é alterado enquanto ele é utilizado, particularmente quando o espaço é diminuído. Tenha extremo cuidado ao ajustar o tamanho do volume.

Embora não seja recomendável, é possível criar um volume esparso especificando o sinalizador `-s` para `zfs create -V` ou alterando a reserva depois que o volume for criado. Um *volume esparso* é um volume em que a reserva não é igual ao tamanho do volume. Em um volume esparso, as alterações para `volsize` não se refletem na reserva.

Para obter mais informações sobre o uso de volumes, consulte [“Volumes ZFS” na página 277](#).

Propriedades de usuário do ZFS

Além das propriedades nativas, o ZFS oferece suporte a propriedades arbitrárias do usuário. As propriedades de usuário não afetam o comportamento do ZFS, mas podem ser usadas para anotar conjuntos de dados com informações significativas no ambiente.

Os nomes das propriedades do usuário devem estar de acordo com as convenções a seguir:

- Eles precisam conter dois pontos (':') para distingui-los das propriedades nativas.
- Eles precisam conter letras minúsculas, números ou caracteres de pontuação: ':', '+', ',', '_'.
- O comprimento máximo de uma propriedade de usuário é de 256 caracteres.

A convenção esperada é que o nome da propriedade esteja dividido nos dois componentes seguintes, mas que este espaço de nome não seja imposto pelo ZFS:

module:property

Ao fazer utilização programática das propriedades do usuário, utilize um nome de domínio DNS reverso para o componente *módulo* dos nomes da propriedade para diminuir a possibilidade de que dois pacotes desenvolvidos independentemente utilizem o mesmo nome de propriedade para fins diferentes. Os nomes de propriedade que começam com `com.sun.` estão reservados para utilização da Oracle Corporation.

O valor da propriedade do usuário deve estar de acordo com as convenções a seguir:

- Eles devem consistir de seqüências arbitrárias que são sempre herdadas e nunca validadas.
- O comprimento máximo do valor da propriedade de usuário é de 1024 caracteres.

Por exemplo:

```
# zfs set dept:users=finance userpool/user1
# zfs set dept:users=general userpool/user2
# zfs set dept:users=itops userpool/user3
```

Todos os comandos que operam em propriedades, tais como `zfs list`, `zfs get`, `zfs set`, etc., podem ser usados para manipular propriedades nativas e de usuário.

Por exemplo:

```
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
```

Para limpar uma propriedade de usuário, use o comando `zfs inherit`. Por exemplo:

```
# zfs inherit -r dept:users userpool
```


Se a propriedade não estiver definida em nenhum conjunto de dados pai, ela é totalmente removida.

Consultando informações sobre o sistema de arquivos ZFS

O comando `zfs list` oferece um amplo mecanismo para exibição e consulta de informações sobre o conjunto de dados. Nesta seção, serão tratadas tanto as consultas básicas quanto as consultas complexas.

Listando informações básicas do ZFS

É possível listar as informações básicas do conjunto de dados usando o comando `zfs list` sem opções. Esse comando exibe os nomes de todos os conjuntos de dados no sistema e os valores de suas propriedades utilizadas, disponíveis, referenciadas e ponto de montagem. Para obter mais informações sobre essas propriedades, consulte [“Introduzindo as propriedades do ZFS” na página 187](#).

Por exemplo:

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
pool                476K  16.5G  21K    /pool
pool/clone          18K   16.5G  18K    /pool/clone
pool/home           296K  16.5G  19K    /pool/home
pool/home/marks     277K  16.5G  277K   /pool/home/marks
pool/home/marks@snap 0      -      277K   -
pool/test           18K   16.5G  18K    /test
```

Também é possível usar este comando para exibir conjuntos de dados específicos proporcionando o nome do banco de dados na linha de comando. Adicionalmente, use a opção `-r` para exibir repetidamente todos os descendentes de tal conjunto de dados. Por exemplo:

```
# zfs list -r pool/home/marks
NAME                USED  AVAIL  REFER  MOUNTPOINT
pool/home/marks     277K  16.5G  277K   /pool/home/marks
pool/home/marks@snap 0      -      277K   -
```

É possível utilizar o comando `zfs list` com o ponto de montagem de um sistema de arquivos. Por exemplo:

```
# zfs list /pool/home/marks
NAME                USED  AVAIL  REFER  MOUNTPOINT
pool/home/marks     277K  16.5G  277K   /pool/home/marks
```

O exemplo a seguir ilustra como exibir informações básicas sobre `pool/home/chua` e todos os seus conjuntos de dados descendentes:

```
# zfs list -r tank/home/chua
NAME                                USED  AVAIL  REFER  MOUNTPOINT
tank/home/chua                      26.0K  4.81G  10.0K  /tank/home/chua
tank/home/chua/projects              16K   4.81G   9.0K   /tank/home/chua/projects
tank/home/chua/projects/fs1          8K   4.81G    8K   /tank/home/chua/projects/fs1
tank/home/chua/projects/fs2          8K   4.81G    8K   /tank/home/chua/projects/fs2
```

Para obter informações adicionais sobre o comando `zfs list`, consulte [zfs\(1M\)](#).

Criando consultas complexas de ZFS

A saída `zfs list` pode ser personalizada utilizando as opções `-o`, `-f` e `-H`.

É possível personalizar a saída do valor da propriedade usando a opção `-o` e uma lista separada por vírgula das propriedades desejadas. É possível abastecer qualquer propriedade de conjunto de dados como um argumento válido. Para obter uma lista de todas as propriedades de conjunto de dados suportadas, consulte “[Introduzindo as propriedades do ZFS](#)” na página 187. Além das propriedades definidas, a lista da opção `-o` também pode conter o nome literal para indicar que a saída deve incluir o nome do conjunto de dados.

O exemplo a seguir utiliza `zfs list` para exibir o nome do conjunto de dados, juntamente com os valores das propriedades `sharenfs` e `mountpoint`.

```
# zfs list -o name,sharenfs,mountpoint
NAME                                SHARENFS  MOUNTPOINT
tank                                off        /tank
tank/home                           on         /tank/home
tank/home/ahrens                     on         /tank/home/ahrens
tank/home/bonwick                    on         /tank/home/bonwick
tank/home/chua                       on         /tank/home/chua
tank/home/eschrock                   on         legacy
tank/home/moore                      on         /tank/home/moore
tank/home/tabriz                     ro         /tank/home/tabriz
```

A opção `-t` pode ser usada para especificar os tipos de conjuntos de dados a serem exibidos. Os tipos válidos estão descritos na tabela abaixo.

TABELA 6-2 Tipos de conjuntos de dados ZFS

Tipo	Descrição
filesystem	Sistemas de arquivos e clones
volume	Volumes
instantâneo	Instantâneos

As opções `-t` utilizam uma lista separada por vírgulas dos tipos de conjuntos de dados a serem exibidos. O exemplo abaixo usa simultaneamente as opções `-t` e `-o` para exibir o nome e a propriedade `used` em todos os sistemas de arquivos:

```
# zfs list -t filesystem -o name,used
NAME                USED
pool                 476K
pool/clone           18K
pool/home            296K
pool/home/marks     277K
pool/test            18K
```

A opção `-H` pode ser usada para omitir o cabeçalho `zfs list` da saída gerada. Com a opção `-H`, todo espaço em branco é substituído pelo caractere Tab. Esta opção pode ser útil quando uma saída analisável é necessária, por exemplo, ao realizar script. O exemplo abaixo ilustra a saída gerada do uso do comando `zfs list` com a opção `-H`:

```
# zfs list -H -o name
pool
pool/clone
pool/home
pool/home/marks
pool/home/marks@snap
pool/test
```

Gerenciando propriedades do ZFS

As propriedades do conjunto de dados são gerenciadas através dos subcomandos `set`, `inherit` e `get` do comando `zfs`.

- “Definindo propriedades do ZFS” na página 203
- “Herdando propriedades do ZFS” na página 204
- “Consultando propriedades do ZFS” na página 205

Definindo propriedades do ZFS

O comando `zfs set` pode ser usado para modificar qualquer propriedade definível de conjunto de dados. Ou então, é possível utilizar o comando `zfs create` para definir as propriedades quando o conjunto de dados é criado. Para obter uma lista de propriedades definíveis de conjunto de dados, consulte “Propriedades nativas definíveis do ZFS” na página 197.

O comando `zfs set` utiliza uma sequência de propriedade/valor no formato de *propriedade=valor* seguido por um nome de conjunto de dados. Somente uma propriedade pode ser configurada ou modificada durante cada chamada `zfs set`.

O exemplo abaixo define a propriedade `atime` como `off` para `tank/home`.

```
# zfs set atime=off tank/home
```

Além disso, qualquer propriedade do sistema de arquivos pode ser definida quando o sistema de arquivos é criado. Por exemplo:

```
# zfs create -o atime=off tank/home
```

É possível especificar valores de propriedades numérica utilizando os sufixos fáceis de entender (em ordem crescente de magnitude): BKMGTPeZ. Todos estes sufixos podem ser seguidos de b opcional, indicando bytes, com exceção do sufixo B, que já indica bytes. As quatro chamadas a seguir de `zfs set` são expressões numéricas equivalentes que configuram a propriedade `quota` para ser definida com o valor de 50 GB no sistema de arquivos `tank/home/marks`:

```
# zfs set quota=50G tank/home/marks
# zfs set quota=50g tank/home/marks
# zfs set quota=50GB tank/home/marks
# zfs set quota=50gb tank/home/marks
```

Os valores de propriedades não numéricas fazem a diferenciação entre maiúsculas e minúsculas e devem ser em minúsculas, com exceção de `mountpoint` e `sharenfs`. Os valores destas propriedades podem apresentar maiúsculas e minúsculas misturadas.

Para obter mais informações sobre o comando `zfs set`, consulte [zfs\(1M\)](#).

Herdando propriedades do ZFS

Todas as propriedades configuráveis, com exceção de cotas e reservas, herdam o valor do seu conjunto de dados pai, a menos que uma cota ou reserva esteja explicitamente definida no conjunto de dados descendente. Se nenhum antepassado tiver um valor explícito definido para uma propriedade herdada, é usado o valor padrão para a propriedade. É possível utilizar o comando `zfs inherit` para limpar um valor de propriedade, fazendo, assim, com que o valor seja herdado do conjunto de dados pai.

O exemplo abaixo usa o comando `zfs set` para ativar a compactação do sistema de arquivos `tank/home/bonwick`. Em seguida, `zfs inherit` é utilizado para limpar a propriedade `compression`, fazendo, assim, com que a propriedade herde o valor padrão de `off`. Como nem `home` e tampouco `tank` possui a propriedade `compression` definida localmente, o valor padrão é utilizado. Se ambos possuírem a compactação ativada, o valor definido no próximo predecessor seria utilizado (neste exemplo `home`).

```
# zfs set compression=on tank/home/bonwick
# zfs get -r compression tank
NAME                PROPERTY    VALUE                SOURCE
tank                 compression off                  default
tank/home            compression off                  default
tank/home/bonwick   compression on                 local
# zfs inherit compression tank/home/bonwick
# zfs get -r compression tank
NAME                PROPERTY    VALUE                SOURCE
tank                 compression off                  default
tank/home            compression off                  default
tank/home/bonwick   compression off                  default
```

O subcomando `inherit` é aplicado repetidamente quando a opção `-r` está especificada. No exemplo abaixo, o comando faz com que o valor da propriedade `compression` seja herdada por `tank/home` e pelos descendentes que possa ter:

```
# zfs inherit -r compression tank/home
```

Observação – Tenha em mente que o uso da opção `-r` limpa a definição da propriedade atual de todos os conjuntos de dados descendentes.

Para mais informações sobre o comando `zfs inherit`, consulte [zfs\(1M\)](#).

Consultando propriedades do ZFS

A forma mais simples de consultar os valores de uma propriedade é usando o comando `zfs list`. Para obter mais informações, consulte [“Listando informações básicas do ZFS” na página 201](#). No entanto, para consultas complexas e para script, use o comando `zfs get` para fornecer informações mais detalhadas em um formato personalizado.

O comando `zfs get` pode ser usado para recuperar qualquer propriedade de conjunto de dados. O exemplo a seguir ilustra como recuperar um valor de propriedade único em um conjunto de dados:

```
# zfs get checksum tank/ws
NAME          PROPERTY      VALUE          SOURCE
tank/ws       checksum      on             default
```

A quarta coluna, `SOURCE`, indica a origem do valor da propriedade. A tabela a seguir define os possíveis valores de origem.

TABELA 6-3 Possíveis valores de `SOURCE` (comando `zfs get`)

Valor de origem	Descrição
<code>default</code>	Esse valor de propriedade nunca foi definido explicitamente para esse conjunto de dados ou para qualquer um dos seus predecessores. O valor padrão desta propriedade está sendo usado.
<code>inherited from nome-do-conjunto-de-dados</code>	Esse valor de propriedade é herdado a partir do conjunto de dados pai especificado em <code>dataset-name</code> .
<code>local</code>	Este valor de propriedade foi explicitamente definido para este conjunto de dados usando <code>zfs set</code> .
<code>temporary</code>	Esse valor de propriedade foi definido utilizando a opção <code>zfs mount -o</code> e é válido somente durante a montagem. Para obter mais informações sobre as propriedades do ponto de montagem, consulte “Usando propriedades de montagem temporárias” na página 212 .

TABELA 6-3 Possíveis valores de SOURCE (comando `zfs get`) (Continuação)

Valor de origem	Descrição
- (none)	Essa propriedade é somente leitura. Seu valor é gerado pelo ZFS.

É possível utilizar a palavra-chave especial `todos` para recuperar todas os valores de propriedade do conjunto de dados. Os exemplos a seguir utilizam a palavra-chave `todos`:

```
# zfs get all tank/home
NAME      PROPERTY          VALUE              SOURCE
tank/home type              filesystem         -
tank/home creation        Tue Jun 29 11:44 2010 -
tank/home used          21K                -
tank/home available    66.9G              -
tank/home referenced    21K                -
tank/home compressratio 1.00x              -
tank/home mounted       yes                 -
tank/home quota         none                default
tank/home reservation   none                default
tank/home recordsize    128K                default
tank/home mountpoint    /tank/home         default
tank/home sharenfs      off                 default
tank/home checksum      on                  default
tank/home compression   off                 default
tank/home atime         on                  default
tank/home devices       on                  default
tank/home exec           on                  default
tank/home setuid        on                  default
tank/home readonly     off                 default
tank/home zoned         off                 default
tank/home snapdir      hidden              default
tank/home aclmode       groupmask           default
tank/home aclinherit    restricted           default
tank/home canmount      on                  default
tank/home shareiscsi    off                 default
tank/home xattr         on                  default
tank/home copies        1                   default
tank/home version       4                   -
tank/home utf8only     off                 -
tank/home normalization none                 -
tank/home casesensitivity sensitive            -
tank/home vscan         off                 default
tank/home nbmand        off                 default
tank/home sharesmb      off                 default
tank/home refquota     none                default
tank/home refreservation none                default
tank/home primarycache  all                 default
tank/home secondarycache all                 default
tank/home usedbysnapshots 0                    -
tank/home usedbydataset  21K                 -
tank/home usedbychildren 0                    -
tank/home usedbyrefreservation 0                    -
tank/home logbias       latency             default
```

Observação – As propriedades `casesensitivity`, `nbmand`, `normalization`, `sharesmb`, `utf8only` e `vsca` não são totalmente operacionais na versão 10 do Oracle Solaris porque o serviço Oracle Solaris SMB não possui suporte na versão 10 do Oracle Solaris.

A opção `-s` de `zfs get` permite especificar, por tipo de origem, o tipo de propriedades a exibir. Esta opção usa uma lista separada por vírgulas indicando os tipos de origem desejados. Somente as propriedades com o tipo de origem especificado são exibidas. Os tipos de origem válidos são `local`, `default`, `inherited`, `temporary` e `none`. O exemplo abaixo ilustra todas as propriedades definidas localmente no pool.

```
# zfs get -s local all pool
NAME          PROPERTY      VALUE          SOURCE
pool          compression   on             local
```

Todas as opções acima podem ser combinadas com a opção `-r` para exibir repetidamente as propriedades especificadas em todos os filhos do conjunto de dados especificado. No exemplo abaixo, todas as propriedades temporárias de todos os conjuntos de dados dentro de `tank` são exibidas repetidamente:

```
# zfs get -r -s temporary all tank
NAME          PROPERTY      VALUE          SOURCE
tank/home     atime         off            temporary
tank/home/bonwick atime         off            temporary
tank/home/marks atime         off            temporary
```

É possível consultar os valores de propriedade utilizando o comando `zfs get` sem especificar um sistema de arquivos destino, o que significa que o comando opera em todos os conjuntos ou sistemas de arquivos. Por exemplo:

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

Para obter mais informações sobre o comando `zfs get`, consulte [zfs\(1M\)](#).

Consultando propriedades do ZFS em busca de script

O comando `zfs get` oferece suporte às opções `-H` e `-o`, que estão destinadas a script. É possível utilizar a opção `-H` para omitir o cabeçalho de informação e substituir o espaço em branco com o caractere `Tab`. O espaço em branco uniforme leva em consideração os dados facilmente analisáveis. É possível utilizar a opção `-o` para personalizar a saída das seguintes maneiras:

- O nome literal pode ser utilizado com uma lista de propriedades separadas por vírgula, como definido na seção [“Introduzindo as propriedades do ZFS” na página 187](#).
- Uma lista de campos literais separados por vírgula, nome, valor, propriedades e origem, a ser retirada seguida por um espaço e um argumento, que é uma lista de propriedades separadas por vírgula.

O exemplo a seguir ilustra como recuperar um único valor utilizando as opções `-H` e `-o` de `zfs get`:

```
# zfs get -H -o value compression tank/home
on
```

A opção `-p` relata valores numéricos como seus valores exatos. Por exemplo, 1 MB seria relatado como 1000000. Esta opção pode ser usada da seguinte forma:

```
# zfs get -H -o value -p used tank/home
182983742
```

É possível utilizar a opção `-r`, juntamente com qualquer uma das opções anteriores, para recuperar recursivamente os valores solicitados para todos os descendentes. O exemplo a seguir utiliza as opções `-H`, `-o` e `-r` para recuperar o nome do conjunto de dados e o valor da propriedade utilizada para `export/home` e seus descendentes, ao mesmo tempo em que omite a saída de cabeçalho:

```
# zfs get -H -o name,value -r used export/home
export/home          5.57G
export/home/marks    1.43G
export/home/maybee   2.15G
```

Montando e compartilhando sistemas de arquivos ZFS

Esta seção descreve como os pontos de montagem e os sistemas de arquivos compartilhados são gerenciados no ZFS.

- “Gerenciando pontos de montagem do ZFS” na página 208
- “Montando sistemas de arquivos ZFS” na página 210
- “Usando propriedades de montagem temporárias” na página 212
- “Desmontando sistemas de arquivos” na página 212
- “Compartilhando e descompartilhando sistemas de arquivos ZFS” na página 213

Gerenciando pontos de montagem do ZFS

Por padrão, o sistema de arquivos ZFS é automaticamente montado quando é criado. É possível determinar comportamento de ponto de montagem específico para um sistema de arquivos como descrito nesta seção.

Também é possível definir o ponto de montagem padrão para o conjunto de dados de um conjunto no momento da criação utilizando, de `zpool create`, a opção `-m`. Para obter mais informações sobre a criação de pools de armazenamento, consulte “Criando um pool de armazenamento do ZFS” na página 72.

Todos os sistemas de arquivos ZFS são montados pelo ZFS no momento da inicialização utilizando o serviço `svc://system/filesystem/local` do Service Management Facility's (SMF). Os sistemas de arquivos são montados em */path*, onde *path* é o nome do sistema de arquivos.

É possível ignorar o ponto de montagem padrão utilizando o comando `zfs set` para configurar a propriedade `mountpoint` para um caminho específico. O ZFS cria automaticamente o ponto de montagem especificado, se necessário, e monta automaticamente o sistema de arquivos associado quando o comando `zfs mount -a` é chamado, sem a necessidade de que o arquivo `/etc/vfstab` seja editado.

A propriedade `mountpoint` é herdada. Por exemplo, se `pool/home` possui a propriedade `mountpoint` definida como `/export/stuff`, então `pool/home/user` herda `/export/stuff/user` para seu valor de propriedade `mountpoint`.

Para prevenir que um sistema de arquivos seja montado, configure a propriedade ponto de montagem para nenhum. Além disso, a propriedade `canmount` pode ser utilizada para controlar se o sistema de arquivos pode ser montado. Para obter mais informações sobre a propriedade `canmount`, consulte [“A propriedade canmount” na página 198](#).

Sistemas de arquivos também podem ser explicitamente gerenciados por interfaces de montagem de legado utilizando `zfs set` para configurar a propriedade ponto de montagem como legado. Fazer isso previne o ZFS da montagem automática e de gerenciar um sistema de arquivos. As ferramentas de legado, que incluem os comandos `mount` e `umount`, e o arquivo `/etc/vfstab` devem ser usados. Para obter mais informações sobre montagem de legado, consulte [“Pontos de montagem de legado” na página 210](#).

Pontos de montagem automáticos

- Ao alterar a propriedade ponto de montagem de legado ou nenhum para um caminho específico, o ZFS monta automaticamente o sistema de arquivos.
- Se o ZFS está gerenciando um sistema de arquivos mas está atualmente desmontado e a propriedade ponto de montagem está alterada, o sistema de arquivos permanece desmontado.

Todo conjunto de dados cuja propriedade `mountpoint` não for `legacy` é gerenciado pelo ZFS. No exemplo a seguir, o conjunto de dados é criado e seu ponto de montagem é automaticamente gerenciado pelo 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                                -
```

Também é possível definir explicitamente a propriedade `mountpoint` conforme ilustrado no exemplo abaixo:

```
# 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                -
```

Quando a propriedade `mountpoint` é alterada, o sistema de arquivos é desmontado automaticamente do ponto de montagem antigo e é montado novamente no novo ponto de montagem. Os diretórios de ponto de montagem são criados conforme necessário. Se o ZFS é incapaz de desmontar um sistema arquivos devido ao fato de estar ativo, um erro é relatado e uma desmontagem manual forçada é necessária.

Pontos de montagem de legado

É possível gerenciar sistemas de arquivos ZFS com ferramentas de legado definindo a propriedade `mountpoint` como `legacy`. Os sistemas de arquivos de legado devem ser gerenciados através dos comandos `mount` e `umount` e do arquivo `/etc/vfstab`. O ZFS não monta automaticamente os sistemas de arquivos de legado no momento da inicialização e os comandos `mount` e `umount` do ZFS não operam em conjuntos de dados desse tipo. Os exemplos abaixo ilustram como configurar e gerenciar um conjunto de dados ZFS no modo de legado:

```
# zfs set mountpoint=legacy tank/home/eschrock
# mount -F zfs tank/home/eschrock /mnt
```

Para montar automaticamente um sistema de arquivos de legado no momento da inicialização, adicione um entrada ao arquivo `/etc/vfstab`. O exemplo a seguir mostra qual entrada no arquivo `/etc/vfstab` poderia se parecer:

```
#device          device          mount          FS      fsck    mount  mount
#to mount        to fsck        point          type    pass   at    boot options
#
tank/home/eschrock -          /mnt          zfs      -       yes   -
```

As entradas `device to fsck` e `fsck pass` são configuradas para `-` porque o comando `fsck` não é aplicável aos sistemas de arquivos ZFS. Para mais informação sobre a integridade de dados ZFS, consulte [“Semânticas transacionais”](#) na página 46.

Montando sistemas de arquivos ZFS

O ZFS monta automaticamente os sistemas de arquivos quando os sistemas de arquivos são criados ou quando o sistema é inicializado. Utilizar o comando `zfs mount` é necessário somente quando for preciso alterar as opções de montagem ou explicitamente montar ou desmontar sistemas de arquivos.

O comando `zfs mount` sem argumentos mostra todos os sistemas de arquivos montados atualmente que são gerenciados pelo ZFS. Os pontos de montagem de legado gerenciados não são exibidos. Por exemplo:

```
# zfs mount
tank                /tank
tank/home           /tank/home
tank/home/bonwick   /tank/home/bonwick
tank/ws             /tank/ws
```

A opção `-a` pode ser usada para montar todos os sistemas de arquivos ZFS gerenciados. Os sistemas de arquivos gerenciados de legado não são montados. Por exemplo:

```
# zfs mount -a
```

Por padrão, o ZFS não permite montagem em cima de um diretório não vazio. Para forçar uma montagem em cima de um diretório não vazio, é necessário usar a opção `-O`. Por exemplo:

```
# zfs mount tank/home/lalt
cannot mount '/export/home/lalt': directory is not empty
use legacy mountpoint to allow this behavior, or use the -O flag
# zfs mount -O tank/home/lalt
```

Pontos de montagem de legado devem ser gerenciados através de ferramentas de legado. Uma tentativa de usar ferramentas do ZFS resulta em erro. Por exemplo:

```
# zfs mount pool/home/billm
cannot mount 'pool/home/billm': legacy mountpoint
use mount(1M) to mount this filesystem
# mount -F zfs tank/home/billm
```

Quando um sistema de arquivos é montado, ele usa um conjunto de opções de montagem com base nos valores de propriedade associados ao conjunto de dados. A correlação entre propriedades e opções de montagem é a seguinte:

TABELA 6-4 Propriedades relacionadas à montagem ZFS e opções de montagem

Propriedade	Opções de montagem
<code>atime</code>	<code>atime/noatime</code>
<code>devices</code>	<code>devices/nodevices</code>
<code>exec</code>	<code>exec/noexec</code>
<code>nbmand</code>	<code>nbmand/nonbmand</code>
<code>readonly</code>	<code>ro/rw</code>
<code>setuid</code>	<code>setuid/nosetuid</code>
<code>xattr</code>	<code>xattr/noaxttr</code>

A opção de montagem `nosuid` é um alias para `nodevices`, `nosetuid`.

Usando propriedades de montagem temporárias

Se qualquer opção de montagem descrita na última seção for explicitamente configurada utilizando a opção `-o` com o comando `zfs mount`, o valor da propriedade associada será temporariamente ignorado. Esses valores de propriedade são relatados como temporários pelo comando `zfs get` e voltam ao valor original quando o sistema de arquivos é desmontado. Se um valor de propriedade for alterado enquanto o conjunto de dados estiver sendo montado, a alteração terá efeito imediatamente, substituindo qualquer definição temporária.

No exemplo a seguir, a opção de montagem de somente leitura está temporariamente configurada no sistema de arquivos `tank/home/perrin`. Presume-se que o sistema de arquivos está desmontado.

```
# zfs mount -o ro tank/home/perrin
```

Para alterar temporariamente um valor de propriedade no sistema de arquivos que esteja atualmente montado, utilize a opção remontar especial. No exemplo abaixo, a propriedade `atime` é temporariamente alterada para `off` para um sistema de arquivos atualmente montado:

```
# zfs mount -o remount,noatime tank/home/perrin
# zfs get atime tank/home/perrin
NAME                PROPERTY    VALUE                SOURCE
tank/home/perrin    atime      off                  temporary
```

Para obter mais informações sobre o comando `zfs mount`, consulte [zfs\(1M\)](#).

Desmontando sistemas de arquivos

É possível desmontar os sistemas de arquivos ZFS utilizando o subcomando `zfs unmount`. O comando `unmount` pode assumir o ponto de montagem ou o nome de sistema de arquivos como um argumento.

No exemplo a seguir, um sistema de arquivos é desmontado pelo seu nome de sistema de arquivos:

```
# zfs unmount tank/home/tabriz
```

No exemplo a seguir, o sistema de arquivos é desmontado pelo seu ponto de montagem:

```
# zfs unmount /export/home/tabriz
```

O comando `unmount` falha se o sistema de arquivos está ocupado. Para desmontar a força um sistema de arquivos, é possível utilizar a opção `-f`. Seja cauteloso ao desmontar a força um sistema de arquivos se os conteúdos estiverem ativamente sendo utilizados. Pode resultar em um comportamento imprevisível do aplicativo.

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

Para proporcionar compatibilidade com versões anteriores, o comando `umount` de legado pode ser utilizado para desmontar sistemas de arquivos ZFS. Por exemplo:

```
# umount /export/home/bob
```

Para obter mais informações sobre o comando `zfs unmount`, consulte [zfs\(1M\)](#).

Compartilhando e descompartilhando sistemas de arquivos ZFS

O ZFS pode compartilhar automaticamente sistemas de arquivos configurando a propriedade `sharenfs`. Utilizando essa propriedade, não é necessário modificar o arquivo `/etc/dfs/dfstab` quando um novo sistema de arquivos é compartilhado. A propriedade `sharenfs` é uma lista de opções separada por vírgulas para passar para o comando `share`. O valor `on` é um alias para opção de compartilhamento padrão, que fornece permissões `read/write` para todos. O valor `off` indica que o sistema de arquivos não é gerenciado pelo ZFS e pode ser compartilhado por meios tradicionais, assim como o arquivo `/etc/dfs/dfstab`. Todos os sistemas de arquivos cuja propriedade `sharenfs` não é `off` são compartilhados durante a inicialização.

Controlando a semântica de compartilhamento

Por padrão, todos os sistemas de arquivos são descompartilhados. Para compartilhar um novo sistema de arquivos, use a sintaxe `zfs set` semelhante à seguinte:

```
# zfs set sharenfs=on tank/home/eschrock
```

A propriedade `sharenfs` é herdada, e os sistemas de arquivos são automaticamente compartilhados na criação se suas propriedades herdadas não forem `off`. Por exemplo:

```
# zfs set sharenfs=on tank/home
# zfs create tank/home/bricker
# zfs create tank/home/tabriz
# zfs set sharenfs=ro tank/home/tabriz
```

`tank/home/bricker` e `tank/home/tabriz` são inicialmente compartilhados como graváveis porque herdam a propriedade `sharenfs` do `tank/home`. Depois que a propriedade é configurada para `ro` (somente leitura), `tank/home/tabriz` é compartilhado como somente leitura independente da propriedade `sharenfs` que é configurada para `tank/home`.

Descompartilhando sistemas de arquivos ZFS

Embora muitos sistemas de arquivos sejam automaticamente compartilhados ou descompartilhados durante a inicialização, criação e destruição, os sistemas de arquivos, algumas vezes, necessitam ser explicitamente descompartilhados. Para isso, use o comando `zfs unshare`. Por exemplo:

```
# zfs unshare tank/home/tabriz
```

Este comando descompartilha o sistema de arquivos `tank/home/tabriz`. Para descompartilhar todos os sistemas de arquivos ZFS no sistema, é necessário usar a opção `-a`.

```
# zfs unshare -a
```

Compartilhando sistemas de arquivos ZFS

Na maioria das vezes, o comportamento automático do ZFS em relação ao compartilhamento do sistema de arquivos na inicialização e criação é suficiente para operações regulares. Se, por algum motivo, você descompartilhar um sistema de arquivos, pode compartilhá-lo novamente com o uso do comando `zfs share`. Por exemplo:

```
# zfs share tank/home/tabriz
```

Você também pode compartilhar todos os sistemas de arquivos ZFS no sistema usando a opção `-a`.

```
# zfs share -a
```

Comportamento de compartilhamento de legado

Se a propriedade `sharenfs` é configurada para `off`, então o ZFS não faz a tentativa de compartilhamento ou de descompartilhamento do sistema de arquivos a qualquer hora. Esse valor permite que o administrador de sistemas de arquivos compartilhe por meios tradicionais, assim como o arquivo `/etc/dfs/dfstab`.

Ao contrário do comando `mount` de legado, os comandos `share` e `unshare` de legado ainda podem funcionar nos sistemas de arquivos ZFS. Como resultado, é possível compartilhar manualmente um sistema de arquivos com opções que diferem da opção da propriedade `sharenfs`. Este modelo administrativo não é recomendável. Escolha gerenciar o compartilhamento NFS completamente por ZFS ou pelo arquivo `/etc/dfs/dfstab`. O modelo administrativo do ZFS foi concebido para ser mais simples e menos trabalhoso do que o modelo tradicional.

Definindo cotas e reservas do ZFS

É possível utilizar a propriedade `cota` para definir um limite na quantidade de espaço em disco que um sistema de arquivos necessita utilizar. Além disso, é possível utilizar a propriedade `reserva` para garantir que uma quantidade especificada de espaço em disco esteja disponível para um sistema de arquivos. Ambas as propriedades se aplicam ao conjunto de dados em que são configuradas e a todos os descendentes desse conjunto de dados.

Isto é, se uma cota é definida no conjunto de dados `tank/home`, a quantidade total do espaço em disco utilizado pelo `tank/home` e *todos os seus descendentes* não podem exceder a cota. Do mesmo modo, se ao `tank/home` é dado uma reserva, `tank/home` e *todos os seus descendentes* utilizarão tal reserva. A quantidade de espaço em disco utilizado por um conjunto de dados e todos os seus descendentes são relatados pela propriedade utilizada.

As propriedades `refquota` e `refreservation` são utilizadas para gerenciar o espaço do sistema de arquivos sem contabilizar o espaço em disco utilizado pelos descendentes, tal como instantâneos e clones.

Nessa versão do Solaris, é possível definir uma cota de *usuário* ou de *grupo* na quantidade de espaço em disco utilizado por arquivos que pertencem a um usuário particular ou a um grupo. As propriedades de cota do usuário ou grupo não podem ser definidas em um volume, em um sistema de arquivos antes da versão 4 do sistema de arquivos, ou em um pool antes da versão 15 do pool.

Considere os pontos a seguir para determinar quais recursos de cotas e reservas melhor ajudam no gerenciamento dos sistemas de arquivos:

- As propriedades `cota` e `reserva` são convenientes para gerenciar o espaço em disco utilizado pelo conjunto de dados e seus descendentes.
- As propriedades `refquota` e `refreservation` são apropriadas para gerenciar o espaço em disco consumido pelo conjunto de dados.
- Configurando as propriedades `refquota` ou `refreservation` superiores às propriedades `cota` ou `reserva` não possuem efeito. Se configurar as propriedades `quota` ou `refquota`, as operações que tentam exceder o valor falham. É possível exceder uma quota que é maior que `refquota`. Por exemplo, se alguns blocos de instantâneos são modificados, é possível exceder a quota antes de exceder `refquota`.
- As cotas do usuário ou grupo fornecem um meio mais fácil de gerenciar o espaço em disco com muitas contas de usuário, como em um ambiente universitário.

Para obter informações sobre a definição de cotas e reservas, consulte [“Definindo cotas em sistemas de arquivos ZFS”](#) na página 216 and [“Definindo reservas nos sistemas de arquivos ZFS”](#) na página 219.

Definindo cotas em sistemas de arquivos ZFS

As cotas nos sistemas de arquivos ZFS podem ser configuradas e exibidas utilizando os comandos `zfs set` e `zfs get`. No exemplo a seguir, uma cota de 10 GB é configurada no `tank/home/bonwick`:

```
# zfs set quota=10G tank/home/bonwick
# zfs get quota tank/home/bonwick
NAME                PROPERTY          VALUE              SOURCE
tank/home/bonwick  quota             10.0G             local
```

As cotas também afetam a saída dos comandos `zfs list` e `df`. Por exemplo:

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home           16.5K 33.5G  8.50K  /export/home
tank/home/bonwick   15.0K 10.0G  8.50K  /export/home/bonwick
tank/home/bonwick/ws 6.50K 10.0G  8.50K  /export/home/bonwick/ws
# df -h /export/home/bonwick
Filesystem          size  used  avail  capacity  Mounted on
tank/home/bonwick   10G   8K   10G    1%        /export/home/bonwick
```

Note que, embora `tank/home` possua 33.5 GB de espaço em disco disponível, `tank/home/bonwick` e `tank/home/bonwick/ws` possuem, cada um, somente 10 GB de espaço em disco disponível, devido à cota em `tank/home/bonwick`.

Não é possível definir uma cota para uma quantidade menor do que a atualmente em uso por um conjunto de dados. Por exemplo:

```
# zfs set quota=10K tank/home/bonwick
cannot set quota for 'tank/home/bonwick': size is less than current used or reserved space
```

É possível definir uma `refquota` em um conjunto de dados que limita quantidade montante de espaço em disco que o conjunto de dados pode consumir. Esse limite de disco rígido não inclui o espaço em disco que é utilizado pelos descendentes. Por exemplo:

```
# zfs set refquota=10g students/studentA
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
profs                106K 33.2G  18K   /profs
students             57.7M 33.2G  19K   /students
students/studentA   57.5M 9.94G  57.5M /students/studentA
# zfs snapshot students/studentA@today
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
profs                106K 33.2G  18K   /profs
students             57.7M 33.2G  19K   /students
students/studentA   57.5M 9.94G  57.5M /students/studentA
students/studentA@today 0      -    57.5M  -
```

Para maior comodidade, é possível configurar outra cota no conjunto de dados para ajudar a gerenciar o espaço em disco que é utilizado pelos instantâneos. Por exemplo:


```
# zfs set quota=20g students/studentA
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
profs                106K  33.2G  18K    /profs
students            57.7M  33.2G  19K    /students
students/studentA   57.5M  9.94G  57.5M  /students/studentA
students/studentA@today  0      -      57.5M  -
```

Nesse cenário, studentA deve alcançar refquota (10 GB) de limite de disco rígido, mas studentA pode remover arquivos para recuperar, mesmo se existirem instantâneos.

No exemplo anterior, a menor das duas cotas (10 GB comparado a 20 GB) é exibida na saída `zfs list`. Para visualizar o valor de ambas as cotas, utilize o comando `zfs get`. Por exemplo:

```
# zfs get refquota,quota students/studentA
NAME                PROPERTY  VALUE      SOURCE
students/studentA  refquota  10G        local
students/studentA  quota     20G        local
```

Definindo usuário e grupo de cotas em um sistema de arquivos ZFS

É possível definir uma cota de usuário ou de grupo utilizando os comandos `zfs userquota` ou `zfs groupquota`, respectivamente. Por exemplo:

```
# zfs create students/compsci
# zfs set userquota@student1=10G students/compsci
# zfs create students/labstaff
# zfs set groupquota@staff=20GB students/labstaff
```

Exiba a cota atual do usuário ou grupo como segue:

```
# zfs get userquota@student1 students/compsci
NAME                PROPERTY  VALUE      SOURCE
students/compsci  userquota@student1  10G        local
# zfs get groupquota@staff students/labstaff
NAME                PROPERTY  VALUE      SOURCE
students/labstaff  groupquota@staff    20G        local
```

É possível exibir o usuário geral ou o grupo de espaço em disco consultando as propriedades a seguir:

```
# zfs userspace students/compsci
TYPE  NAME  USED  QUOTA
POSIX User  root  227M  none
POSIX User  student1  455M  10G
# zfs groupspace students/labstaff
TYPE  NAME  USED  QUOTA
POSIX Group  root  217M  none
POSIX Group  staff  217M  20G
```

Para identificar usuários individuais ou de grupo de uso do espaço em disco, consulte as propriedades a seguir:

```
# zfs get userused@student1 students/compsci
NAME          PROPERTY      VALUE          SOURCE
students/compsci userused@student1 455M          local
# zfs get groupused@staff students/labstaff
NAME          PROPERTY      VALUE          SOURCE
students/labstaff groupused@staff 217M          local
```

As propriedades de cotas de usuário e de grupo não são exibidas utilizando o comando `zfs get all conjunto de dados`, que exibem uma lista de todas as outras propriedades do sistema de arquivos.

É possível remover uma cota de usuário de grupo da seguinte forma:

```
# zfs set userquota@user1=none students/compsci
# zfs set groupquota@staff=none students/labstaff
```

As cotas de usuário e de grupo no sistema de arquivos ZFS fornecem os recursos a seguir:

- Uma cota de usuário ou de grupo definida no sistema de arquivos pai que não é automaticamente herdada por um sistema de arquivos descendente.
- No entanto, a cota do usuário ou grupo é aplicada quando um clone ou um instantâneo é criado a partir de um sistema de arquivos que tenha uma cota do usuário ou grupo. Do mesmo modo, uma cota de usuário ou de grupo que está incluída com o sistema de arquivos quando um fluxo é criado utilizando o comando `zfs send`, mesmo sem a opção `-R`.
- Usuários sem privilégio só podem acessar seu próprio espaço em disco. O usuário raiz ou um usuário a quem foi concedido o privilégio de `userused` ou `groupused`, pode acessar a informação de contabilidade do espaço em disco de um grupo ou usuário de todos.
- As propriedades `userquota` e `groupquota` não podem ser definidas nos volumes ZFS, em um sistema de arquivos anterior a versão 4, ou em um pool anterior a versão 15 do pool.

Aplicação de usuário e cotas de grupos podem ser adiadas por vários segundos. Esta atraso significa que os usuários podem exceder suas cotas antes que o sistema observe que estão acima da cota e recusa gravações adicionais `EDQUOT`.

Você pode usar o comando `quota` herdado, para rever as cotas do usuário em um ambiente NFS, por exemplo, onde uma sistema de arquivos ZFS não esteja montado. Sem quaisquer opções, o comando `quota` somente exibe a saída se a cota do usuário estiver excedida. Por exemplo:

```
# zfs set userquota@student1=10m students/compsci
# zfs userspace students/compsci
TYPE      NAME      USED  QUOTA
POSIX User root      227M none
POSIX User student1 455M 10M
# quota student1
Block limit reached on /students/compsci
```

Se você predefinir a cota de usuário e o limite de cotas não estiver excedido, é possível utilizar o comando `quota -v` para visualizar a cota do usuário. Por exemplo:

```
# zfs set userquota@student1=10GB students/compsci
# zfs userspace students/compsci
TYPE      NAME      USED  QUOTA
POSIX User root      227M  none
POSIX User student1 455M  10G
# quota student1
# quota -v student1
Disk quotas for student1 (uid 201):
Filesystem      usage quota limit   timeleft files quota limit   timeleft
/students/compsci
466029 10485760 10485760
```

Definindo reservas nos sistemas de arquivos ZFS

Uma *reserva* ZFS é uma alocação do espaço em disco do conjunto que garante estar disponível ao conjunto de dados. Assim, não é possível reservar o espaço em disco para um conjunto de dados se o espaço não está atualmente disponível no conjunto. A quantidade total de reservas livres e não consumidas não pode exceder a quantidade de espaço em disco não utilizado no conjunto. As reservas do ZFS podem ser definidas e exibidas com os comandos `zfs set` e `zfs get`. Por exemplo:

```
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME          PROPERTY  VALUE  SOURCE
tank/home/moore reservation 5G     local
```

As reservas podem afetar a saída do comando `zfs list`. Por exemplo:

```
# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
tank/home     5.00G 33.5G 8.50K  /export/home
tank/home/moore 15.0K 33.5G 8.50K  /export/home/moore
```

Note que `tank/home` está utilizando 5 GB de espaço em disco, embora a quantidade total do espaço seja referente a `tank/home` e seus descendentes são menores que 5 GB. O espaço usado reflete o espaço reservado para `tank/home/moore`. As reservas são consideradas no cálculo de espaço em disco utilizado do conjunto de dados pai e contam contra sua cota, reserva ou ambos.

```
# 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
```

Um conjunto de dados utiliza mais espaço em disco do que sua reserva, enquanto o espaço não reservado está disponível no conjunto e o conjunto de dados atualmente utilizado está abaixo de sua cota. Um conjunto de dados não pode utilizar o espaço em disco que foi reservado para outro conjunto de dados.

As reservas não são cumulativas. Ou seja, uma segunda chamada de `zfs set` para definir uma reserva não adiciona sua reserva à reserva existente. Em vez disso, a segunda reserva substitui a primeira reserva. Por exemplo:

```
# zfs set reservation=10G tank/home/moore
# zfs set reservation=5G tank/home/moore
# zfs get reservation tank/home/moore
NAME                PROPERTY      VALUE          SOURCE
tank/home/moore     reservation   5.00G         local
```

É possível definir uma reserva `reservation` para garantir espaço em disco para um conjunto de dados que não inclua espaço em disco utilizado pelos instantâneos e clones. Essa reserva é contabilizada para o cálculo do espaço utilizado do conjunto de dados pai e conta contra as cotas e reservas do conjunto de dados pai. Por exemplo:

```
# 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
```

Também é possível definir um reserva no mesmo conjunto de dados para garantir o espaço do conjunto de dados e o espaço do instantâneo. Por exemplo:

```
# 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
```

As reservas regulares são contabilizados para o cálculo de espaço utilizado do pai.

No exemplo anterior, a menor das duas cotas (10 GB comparado a 20 GB) é exibida na saída `zfs list`. Para visualizar o valor de ambas as cotas, utilize o comando `zfs get`. Por exemplo:

```
# zfs get reservation,refreserv profs/prof1
NAME                PROPERTY      VALUE          SOURCE
profs/prof1         reservation   20G            local
profs/prof1         refreservation 10G            local
```

Se `refreservation` está configurada, um instantâneo só é permitido se o espaço do conjunto não reservado suficiente existe fora da reserva para acomodar o número atual de bytes *referenciados* no conjunto de dados.

Trabalhando com instantâneos e clones do ZFS do Oracle Solaris

Este capítulo descreve como criar e gerenciar instantâneos e clones do ZFS do Oracle Solaris. Também são fornecidas informações sobre como salvar instantâneos.

Este capítulo traz as seguintes seções:

- “Visão geral dos instantâneos do ZFS” na página 221
- “Criando e destruindo instantâneos do ZFS” na página 222
- “Exibindo e acessando instantâneos do ZFS” na página 225
- “Retornando um instantâneo ZFS” na página 227
- “Visão geral dos clones do ZFS” na página 228
- “Criando um clone do ZFS” na página 228
- “Destruindo um clone do ZFS” na página 229
- “Substituindo um sistema de arquivos ZFS por um clone do ZFS” na página 229
- “Enviando e recebendo dados do ZFS” na página 230

Visão geral dos instantâneos do ZFS

Um *instantâneo* é uma cópia de somente leitura de um sistema de arquivos ou volume. Os instantâneos podem ser criados quase que instantaneamente e, no início, não utilizam espaço adicional em disco dentro do conjunto. No entanto, à medida em que os dados são alterados dentro do conjunto de dados ativo, o instantâneo passa a consumir espaço em disco devido às contínuas referências aos dados antigos e impede, conseqüentemente, a liberação de espaço.

Os instantâneos do ZFS contam com os seguintes recursos:

- Resistência a reinicializações do sistema.
- O número máximo de instantâneos é teoricamente 2^{64} .
- Os instantâneos não utilizam armazenamento de apoio separado. Os instantâneos utilizam espaço em disco diretamente do mesmo conjunto de armazenamento que o sistema de arquivos ou volume a partir do qual foram criados.

- Os instantâneos recursivos são criados rapidamente como uma operação atômica. Os instantâneos são criados juntos (todos de uma vez) ou simplesmente não são criados. O benefício das operações de instantâneos atômicos é que os dados do instantâneo são sempre retirados em um momento consistente, mesmo nos sistemas de arquivos descendentes.

Os instantâneos de volumes não podem ser acessados diretamente, mas podem ser clonados, revertidos, ter um backup, e assim por diante. Para obter mais informações sobre o backup de um instantâneo do ZFS, consulte [“Enviando e recebendo dados do ZFS” na página 230](#).

- [“Criando e destruindo instantâneos do ZFS” na página 222](#)
- [“Exibindo e acessando instantâneos do ZFS” na página 225](#)
- [“Retornando um instantâneo ZFS” na página 227](#)

Criando e destruindo instantâneos do ZFS

Os instantâneos são criados com o comando `zfs snapshot`, que apresenta o nome do instantâneo a ser criado como seu único argumento. O nome do instantâneo é especificado da seguinte forma:

```
filesystem@snapname
volume@snapname
```

O nome do instantâneo deve estar de acordo com os requisitos de identificação em [“Requisitos para nomeação de componentes do ZFS” na página 50](#).

No exemplo seguinte, é criado um instantâneo de `tank/home/ahrens` denominado `friday`.

```
# zfs snapshot tank/home/ahrens@friday
```

É possível criar instantâneos para todos os sistemas de arquivos descendentes com a opção `-r`. Por exemplo:

```
# zfs snapshot -r tank/home@now
# zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool/ROOT/zfs2BE@zfs2BE           78.3M  -      4.53G  -
tank/home@now                       0      -      26K    -
tank/home/ahrens@now                 0      -      259M   -
tank/home/anne@now                   0      -      156M   -
tank/home/bob@now                    0      -      156M   -
tank/home/cindys@now                 0      -      104M   -
```

Os instantâneos não possuem propriedades que possam ser modificadas. E as propriedades do conjunto de dados também não podem ser aplicadas a um instantâneo. Por exemplo:

```
# zfs set compression=on tank/home/ahrens@now
cannot set compression property for 'tank/home/ahrens@now': snapshot
properties cannot be modified
```

Os instantâneos são destruídos com o comando `zfs destroy`. Por exemplo:

```
# zfs destroy tank/home/ahrens@now
```

Um conjunto de dados não pode ser destruído se existirem instantâneos deste conjunto de dados. Por exemplo:

```
# zfs destroy tank/home/ahrens
cannot destroy 'tank/home/ahrens': filesystem has children
use '-r' to destroy the following datasets:
tank/home/ahrens@tuesday
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
```

Além disso, se foram criados clones de um instantâneo, tais clones devem ser destruídos antes que o instantâneo seja destruído.

Para obter mais informações sobre o subcomando `destroy`, consulte [“Destruindo um sistema de arquivos ZFS” na página 185](#).

Contendo instantâneos do ZFS

Se você possui uma política de instantâneo automático diferente, na qual instantâneos mais antigos estão sendo destruídos sem aviso pelo `zfs receive` por não existirem mais no lado de envio, é possível utilizar a função de contenção de instantâneos.

Conter um instantâneo evita que ele seja destruído. Além disso, este recurso permite que instantâneos com clones possam ser excluídos enquanto a remoção do último clone está pendente utilizando o comando `zfs destroy -d`. Cada instantâneo possui uma contagem associada para referência do usuário, inicializada no zero. Esta contagem aumenta de um em um quando uma contenção é colocada em um instantâneo e diminui de um em um quando uma contenção é liberada.

Na versão anterior do Solaris, instantâneos poderiam ser destruídos apenas através do comando `zfs destroy` se esses não tivessem clones. Nesta versão do Solaris, o instantâneo também deve possuir uma contagem para referência do usuário em zero.

É possível conter um instantâneo ou conjunto de instantâneos. Por exemplo, a sintaxe a seguir põe uma marcação para conter, manter, em `tank/home/cindys/snap1`.

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

É possível utilizar a opção `-r` como recurso para conter os instantâneos de todos os sistemas de arquivos descendentes. Por exemplo:

```
# zfs snapshot -r tank/home@now
# zfs hold -r keep tank/home@now
```

Esta sintaxe adiciona uma única referência, `manter`, ao instantâneo ou ao conjunto de instantâneos. Cada instantâneo possui sua própria marcação do espaço de nome e as marcações

de contenção devem ser únicas no espaço. Se existir uma contenção em um instantâneo, as tentativas para destruir o instantâneo contido através do comando `zfs destroy` falharão. Por exemplo:

```
# zfs destroy tank/home/cindys@snap1
cannot destroy 'tank/home/cindys@snap1': dataset is busy
```

Se deseja destruir um instantâneo contido, utilize a opção `-d`. Por exemplo:

```
# zfs destroy -d tank/home/cindys@snap1
```

Utilize o comando `zfs holds` para exibir uma lista de instantâneos contidos. Por exemplo:

```
# zfs holds tank/home@now
NAME          TAG      TIMESTAMP
tank/home@now keep    Thu Jul 15 11:25:39 2010
```

```
# zfs holds -r tank/home@now
NAME          TAG      TIMESTAMP
tank/home/cindys@now keep    Thu Jul 15 11:25:39 2010
tank/home/mark@now keep    Thu Jul 15 11:25:39 2010
tank/home@now keep    Thu Jul 15 11:25:39 2010
```

É possível utilizar o comando `zfs release` para liberar a contenção de um instantâneo ou de um conjunto de instantâneos. Por exemplo:

```
# zfs release -r keep tank/home@now
```

Se uma contenção for liberada, o instantâneo pode ser destruído através do comando `zfs destroy`. Por exemplo:

```
# zfs destroy -r tank/home@now
```

Dois novas propriedades identificam as informações de contenção de instantâneos:

- A propriedade `defer_destroy` está on se o instantâneo foi marcado para destruição adiada utilizando o comando `zfs destroy -d`. Caso contrário, a propriedade está off.
- A propriedade `user_refs` está definida para o número de contenções do instantâneo, também chamada de contagem para referência do usuário.

Renomeando instantâneos do ZFS

É possível renomear instantâneos, porém eles devem ser renomeados dentro do conjunto e do conjunto de dados a partir do qual foram criados. Por exemplo:

```
# zfs rename tank/home/cindys@083006 tank/home/cindys@today
```

Além disso, a sintaxe do atalho a seguir é equivalente à sintaxe anterior:

```
# zfs rename tank/home/cindys@083006 today
```


A operação rename de instantâneo a seguir não é suportada porque o conjunto de destino e o nome do sistema de arquivos são diferentes do conjunto e do sistema de arquivos onde o instantâneo foi criado:

```
# zfs rename tank/home/cindys@today pool/home/cindys@aturday
cannot rename to 'pool/home/cindys@today': snapshots must be part of same
dataset
```

Como recurso, é possível renomear instantâneos com o comando `zfs rename -r`. Por exemplo:

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
users                                270K  16.5G  22K    /users
users/home                           76K   16.5G  22K    /users/home
users/home@yesterday                  0     -     22K    -
users/home/markm                      18K   16.5G  18K    /users/home/markm
users/home/markm@yesterday            0     -     18K    -
users/home/marks                      18K   16.5G  18K    /users/home/marks
users/home/marks@yesterday            0     -     18K    -
users/home/neil                      18K   16.5G  18K    /users/home/neil
users/home/neil@yesterday             0     -     18K    -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -r users/home
NAME                                USED  AVAIL  REFER  MOUNTPOINT
users/home                           76K   16.5G  22K    /users/home
users/home@2daysago                  0     -     22K    -
users/home/markm                      18K   16.5G  18K    /users/home/markm
users/home/markm@2daysago            0     -     18K    -
users/home/marks                      18K   16.5G  18K    /users/home/marks
users/home/marks@2daysago            0     -     18K    -
users/home/neil                      18K   16.5G  18K    /users/home/neil
users/home/neil@2daysago             0     -     18K    -
```

Exibindo e acessando instantâneos do ZFS

Você pode ativar ou desativar a exibição das listas de instantâneos na saída `zfs list` usando a propriedade `listsnapshots` do pool. Esta propriedade está ativada por padrão.

Se desativar esta propriedade, você pode usar o comando `zfs list -t snapshot` para exibir as informações do instantâneo. Ou, ative a propriedade `listsnapshots` do pool. Por exemplo:

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

Os instantâneos de sistemas de arquivos podem ser acessados no diretório `.zfs/snapshot` dentro da raiz do sistema de arquivos. Por exemplo, se `tank/home/ahrens` estiver montado em

/home/ahrens, então os dados do instantâneo tank/home/ahrens@thursday poderão ser acessados no diretório /home/ahrens/.zfs/snapshot/thursday .

```
# ls /tank/home/ahrens/.zfs/snapshot
tuesday wednesday thursday
```

Os instantâneos podem ser listados da seguinte forma:

```
# zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
pool/home/anne@monday              0    -    780K  -
pool/home/bob@monday               0    -    1.01M  -
tank/home/ahrens@tuesday          8.50K  -    780K  -
tank/home/ahrens@wednesday       8.50K  -    1.01M  -
tank/home/ahrens@thursday         0    -    1.77M  -
tank/home/cindys@today            8.50K  -    524K  -
```

Os instantâneos criados para um sistema de arquivos específico podem ser listados da seguinte forma:

```
# zfs list -r -t snapshot -o name,creation tank/home
NAME                                CREATION
tank/home@now                      Wed Jun 30 16:16 2010
tank/home/ahrens@now              Wed Jun 30 16:16 2010
tank/home/anne@now               Wed Jun 30 16:16 2010
tank/home/bob@now                Wed Jun 30 16:16 2010
tank/home/cindys@now             Wed Jun 30 16:16 2010
```

Contabilidade de espaço em disco para instantâneos do ZFS

Quando um instantâneo é criado, seu espaço em disco é inicialmente compartilhado entre o instantâneo e o sistema de arquivos e, possivelmente, com os instantâneos anteriores. À medida que o sistema de arquivos é alterado, o espaço em disco previamente compartilhado torna-se exclusivo do instantâneo e é incluído na propriedade `used` do instantâneo. Além disso, a exclusão de instantâneos pode aumentar a quantidade de espaço em disco exclusivo de (e conseqüentemente ser *utilizado* por) outros instantâneos.

A propriedade `referenced` de espaço do instantâneo é a mesma que o sistema de arquivos tinha no momento em que o instantâneo foi criado.

É possível identificar informações adicionais sobre como os valores da propriedade `used` são consumidos. Novas propriedades de sistema de arquivos de somente leitura descrevem a utilização do espaço em disco para clones, sistemas de arquivos e volumes. Por exemplo:

```
$ zfs list -o space
# zfs list -ro space tank/home
NAME                                AVAIL  USED  USED SNAP  USED DDS  USED REFRESERV  USED CHILD
tank/home                          66.3G  675M  0          26K      0              675M
tank/home@now                       -      0      -          -        -              -
tank/home/ahrens                    66.3G  259M  0          259M    0              0
tank/home/ahrens@now                 -      0      -          -        -              -
```

tank/home/anne	66.3G	156M	0	156M	0	0
tank/home/anne@now	-	0	-	-	-	-
tank/home/bob	66.3G	156M	0	156M	0	0
tank/home/bob@now	-	0	-	-	-	-
tank/home/cindys	66.3G	104M	0	104M	0	0
tank/home/cindys@now	-	0	-	-	-	-

Para obter uma descrição dessas propriedades, consulte a [Tabela 6-1](#).

Retornando um instantâneo ZFS

É possível utilizar o comando `zfs rollback` para descartar todas as alterações feitas ao sistema de arquivos desde a criação de um instantâneo específico. O sistema de arquivos volta ao estado que se encontrava no momento em que o instantâneo foi realizado. Por padrão, o comando só pode reverter um instantâneo ao instantâneo mais recente.

Para reverter a um instantâneo anterior, todos os instantâneos intermediários devem ser destruídos. Os instantâneos anteriores podem ser destruídos especificando a opção `-r`.

Se os instantâneos intermediários possuírem clones, a opção `-R` também deve ser especificada para destruí-los.

Observação – Se o sistema de arquivos que deseja reverter estiver atualmente montado, ele será desmontado e remontado. Se o sistema de arquivos não puder ser desmontado, a reversão falhará. A opção `-f` obriga o sistema de arquivos a ser desmontado, se necessário.

No exemplo abaixo, o sistema de arquivos `tank/home/ahrens` é revertido ao instantâneo `tuesday`:

```
# zfs rollback tank/home/ahrens@tuesday
cannot rollback to 'tank/home/ahrens@tuesday': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
tank/home/ahrens@wednesday
tank/home/ahrens@thursday
# zfs rollback -r tank/home/ahrens@tuesday
```

Neste exemplo, os instantâneos `wednesday` e `thursday` são destruídos porque você reverteu para o instantâneo `tuesday`.

```
# zfs list -r -t snapshot -o name,creation tank/home/ahrens
NAME                CREATION
tank/home/ahrens@now Wed Jun 30 16:16 2010
```

Visão geral dos clones do ZFS

Um *clone* é um volume ou um sistema de arquivos gravável cujo conteúdo inicial é o mesmo do conjunto de dados do qual foi criado. Da mesma forma que os instantâneos, a criação de clones é quase instantânea e inicialmente não utiliza espaço em disco adicional. Além disso, você pode realizar o instantâneo de um clone.

Os clones podem ser criados somente a partir de um instantâneo. Quando um instantâneo é clonado, cria-se uma dependência implícita entre o clone e o instantâneo. Mesmo que o clone tenha sido criado em qualquer parte da hierarquia do conjunto de dados, o instantâneo original não poderá ser destruído enquanto tal clone existir. A propriedade `origin` revela essa dependência e o comando `zfs destroy` lista tais dependências, se existirem.

Os clones não herdam as propriedades do conjunto de dados do qual foram criados. Use os comandos `zfs get` e `zfs set` para ver e alterar as propriedades de um conjunto de dados clonado. Para obter mais informações sobre a configuração das propriedades do conjunto de dados do ZFS, consulte [“Definindo propriedades do ZFS” na página 203](#).

O clone compartilha inicialmente todo o seu espaço em disco com o instantâneo original e, por essa razão, o valor de sua propriedade `used` é inicialmente zero. O clone passa a utilizar mais espaço em disco à medida que sofre alterações. A propriedade `used` do instantâneo original não inclui o espaço em disco consumido pelo clone.

- [“Criando um clone do ZFS” na página 228](#)
- [“Destruindo um clone do ZFS” na página 229](#)
- [“Substituindo um sistema de arquivos ZFS por um clone do ZFS” na página 229](#)

Criando um clone do ZFS

Para criar um clone, use o comando `zfs clone`, especificando o instantâneo a partir do qual criar o clone e o nome do novo sistema de arquivos ou volume. O novo sistema de arquivos ou volume pode ser colocado em qualquer parte da hierarquia do ZFS. O novo conjunto de dados é do mesmo tipo (por exemplo, sistema de arquivos ou volume) que o instantâneo a partir do qual o clone foi criado. É possível criar um clone de um sistema de arquivos em um conjunto diferente daquele no qual se encontra o instantâneo do sistema de arquivos original.

No exemplo a seguir, um novo clone chamado `tank/home/ahrens/bug123` com o mesmo conteúdo inicial do instantâneo `tank/ws/gate@yesterday` é criado:

```
# zfs snapshot tank/ws/gate@yesterday
# zfs clone tank/ws/gate@yesterday tank/home/ahrens/bug123
```

No exemplo abaixo, a partir do instantâneo `projects/newproject@today` é criada, para um usuário temporário, uma área de trabalho como `projects/teamA/tempuser`. As propriedades são, então, definidas na área de trabalho clonada.

```
# zfs snapshot projects/newproject@today
# zfs clone projects/newproject@today projects/teamA/tempuser
# zfs set sharenfs=on projects/teamA/tempuser
# zfs set quota=5G projects/teamA/tempuser
```

Destruindo um clone do ZFS

Os clones do ZFS são eliminados com o comando `zfs destroy`. Por exemplo:

```
# zfs destroy tank/home/ahrens/bug123
```

Os clones devem ser destruídos antes da destruição dos instantâneos pais.

Substituindo um sistema de arquivos ZFS por um clone do ZFS

O comando `zfs promote` pode ser usado para substituir um sistema de arquivos ZFS ativo por um clone de um sistema de arquivos. Este recurso permite clonar e substituir os sistemas de arquivos para que o sistema de arquivos *original* torne-se o clone do sistema de arquivos especificado. Além disso, este recurso possibilita destruir o sistema de arquivos a partir do qual o clone foi originalmente criado. Sem promover o clone, não é possível destruir um sistema de arquivos original de clones ativos. Para obter mais informações sobre a destruição de clones, consulte [“Destruindo um clone do ZFS” na página 229](#).

No exemplo a seguir, o sistema de arquivos `tank/test/productA` é clonado e, então, o clone do sistema de arquivos, `tank/test/productAbeta`, torna-se o sistema de arquivos `tank/test/productA` original.

```
# 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	-

Nesta saída `zfs list`, note que as informações de contabilidade de espaço em disco do sistema de arquivos `productA` original foram substituídas pelo sistema de arquivos `productAbeta`.

É possível completar o processo de substituição de clone renomeando os sistemas de arquivos. Por exemplo:

```
# zfs rename tank/test/productA tank/test/productAlegacy
# zfs rename tank/test/productAbeta tank/test/productA
# zfs list -r tank/test
```

Opcionalmente, você pode remover o sistema de arquivos de legado. Por exemplo:

```
# zfs destroy tank/test/productAlegacy
```

Enviando e recebendo dados do ZFS

O comando `zfs send` cria uma representação do fluxo de um instantâneo gravado em uma saída padrão. Por padrão, um fluxo inteiro é gerado. Você pode redirecionar a saída para um arquivo ou para outro sistema. O comando `zfs receive` cria um instantâneo cujo conteúdo está especificado no fluxo fornecido na entrada padrão. Se um fluxo inteiro for recebido, um novo sistema de arquivos também é criado. Com estes comandos, é possível enviar os dados do instantâneo do ZFS e receber os dados do instantâneo e os sistemas de arquivos ZFS. Consulte os exemplos nesta seção.

- [“Enviando um instantâneo do ZFS” na página 231](#)
- [“Recebendo um instantâneo do ZFS” na página 232](#)
- [“Enviando e recebendo fluxos complexos de instantâneos do ZFS” na página 233](#)
- [“Salvando dados do ZFS com outros produtos de backup” na página 231](#)

Estão disponíveis as seguintes soluções de backup para salvamento de dados do ZFS:

- **Produtos de backup da empresa:** se precisar dos recursos a seguir, então considere uma solução de backup de empresa:
 - Restauração por arquivo
 - Verificação da mídia de backup
 - Gerenciamento de mídia
- **Instantâneos de sistemas de arquivos e reversão de instantâneos:** utilize os comandos `zfs snapshot` e `zfs rollback` se desejar criar facilmente uma cópia de um sistema de arquivos e reverter para uma versão anterior do sistema de arquivos, se necessário. Por exemplo, para restaurar um arquivo ou arquivos de uma versão anterior de um sistema de arquivos, você poderia utilizar esta solução.

Para obter mais informações sobre a criação e reversão de um instantâneo, consulte [“Visão geral dos instantâneos do ZFS” na página 221](#).

- **Salvar instantâneos** – Use os comandos `zfs send` e `zfs receive` para enviar e receber um instantâneo do ZFS. Você pode salvar alterações incrementais entre instantâneos, mas não pode restaurar os arquivos individualmente. Você deve restaurar o instantâneo de todo o sistema de arquivos. Estes comandos não oferecem uma solução de backup completa para salvar dados do ZFS.

- **Replicação remota:** utilize os comandos `zfs send` e `zfs receive` para copiar um sistema de arquivos de um sistema a outro. Este processo difere do produto de gerenciamento de volumes que pode espelhar os dispositivos através de uma WAN. Não é necessário nenhum hardware ou configuração especial. A vantagem da replicação de um sistema de arquivos ZFS é que você pode recriar um sistema de arquivos em um pool de armazenamento de outro sistema e especificar diferentes níveis de configuração para o pool recém-criado, tal como RAID-Z, mas com os dados de sistema de arquivos idênticos.
- Utilitários de arquivo – Salvar dados do ZFS com utilitários de arquivos como `tar`, `cpio` e `pax` ou produtos de backup de terceiros. Atualmente, os comandos `tar` e `cpio` traduzem as ACLs de estilo NFSv4 corretamente, mas o comando `pax` não.

Salvando dados do ZFS com outros produtos de backup

Além dos comandos `zfs send` e `zfs receive`, você também pode usar utilitários de arquivo, como os comandos `tar` e `cpio`, para salvar os arquivos ZFS. Estes utilitários salvam e restauram ACLs e atributos do arquivo do ZFS. Verifique as opções adequadas para os comandos `tar` e `cpio`.

Para obter informações atualizadas sobre problemas com o ZFS e produtos de backup de terceiros, consulte as notas da versão do Solaris 10 ou o FAQ do ZFS, disponível aqui:

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

Enviando um instantâneo do ZFS

Você pode utilizar o comando `zfs send` para enviar uma cópia de um fluxo de instantâneo e receber o fluxo de instantâneo em outro conjunto do mesmo sistema ou em outro conjunto de outro sistema utilizado para armazenar dados de backup. Por exemplo, para enviar o fluxo de instantâneo de outro conjunto do mesmo sistema, utilize uma sintaxe semelhante à seguinte:

```
# zfs send tank/data@snap1 | zfs recv spool/ds01
```

O `zfs recv` pode ser usado como um alias do comando `zfs receive`.

Se estiver enviando o fluxo de instantâneo a outro sistema, conduza a saída de `zfs send` através do comando `ssh`. Por exemplo:

```
host1# zfs send tank/dana@snap1 | ssh host2 zfs recv newtank/dana
```

Ao enviar um fluxo completo, o sistema de arquivos de destino não deve existir.

É possível salvar dados incrementais usando a opção `i` do `-zfs send`. Por exemplo:

```
host1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh host2 zfs recv newtank/dana
```

Observe que o primeiro argumento (*snap1*) é o instantâneo mais antigo e o segundo (*snap2*) é o instantâneo mais recente. Nesse caso, o sistema de arquivos *newtank/dana* deve existir para que o recebimento incremental tenha êxito.

A origem do *snap1* incremental pode ser especificada como o último componente do nome do instantâneo. Este atalho significa que é necessário especificar o nome do *snap1* somente depois do símbolo @, que se supõe ser do mesmo sistema de arquivos que o *snap2*. Por exemplo:

```
host1# zfs send -i snap1 tank/dana@snap2 > ssh host2 zfs recv newtank/dana
```

Esta sintaxe de atalho é equivalente à sintaxe incremental no exemplo anterior.

Se você tentar gerar um fluxo incremental do *instantâneo1* de um sistema de arquivos diferente, é exibida a seguinte mensagem:

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Se precisar restaurar várias cópias, considere compactar uma representação de fluxo de instantâneo do ZFS com o comando *gzip*. Por exemplo:

```
# zfs send pool/fs@snap | gzip > backupfile.gz
```

Recebendo um instantâneo do ZFS

Leve em consideração as seguintes questões ao receber um instantâneo de um sistema de arquivos:

- O instantâneo e o sistema de arquivos são recebidos.
- O sistema de arquivos e todos os sistemas de arquivos descendentes são desmontados.
- Os sistemas de arquivos não podem ser acessados enquanto estiverem sendo recebidos.
- O sistema de arquivos original que será recebido não deve existir enquanto estiver sendo recebido.
- Se um sistema de arquivos com o mesmo nome já existir, é possível utilizar o comando *zfs rename* para renomear o sistema de arquivos.

Por exemplo:

```
# 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
```

Se fizer uma alteração no sistema de arquivos de destino, e quiser efetuar outro envio incremental de um instantâneo, é necessário, primeiramente, reverter o sistema de arquivos receptor.

Considere o seguinte exemplo. Primeiro, faça uma alteração ao sistema de arquivos, como segue:

```
host2# rm newtank/dana/file.1
```

Então, efetue um envio incremental de tank/dana@snap3. No entanto, é necessário reverter o sistema de arquivos receptor para receber o novo instantâneo incremental. Ou então, a etapa de reversão pode ser eliminada utilizando a opção `-F`. Por exemplo:

```
host1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh host2 zfs recv -F newtank/dana
```

Quando um instantâneo incremental é recebido, o sistema de arquivos de destino já deve existir.

Se fizer alterações ao sistema de arquivos e não reverter o sistema de arquivos receptores para receber o novo instantâneo incremental ou não utilizar a opção `-F`, será exibida uma mensagem similar à seguinte:

```
host1# zfs send -i tank/dana@snap4 tank/dana@snap5 | ssh host2 zfs recv newtank/dana
cannot receive: destination has been modified since most recent snapshot
```

As verificações seguintes são realizadas antes que a opção `-F` seja realizada:

- Se o instantâneo mais recente não corresponder à fonte incremental, nem a reversão nem o recebimento são concluídos e uma mensagem de erro é devolvida.
- Se acidentalmente for fornecido o nome de um sistema de arquivos diferentes que não correspondem à fonte incremental especificada no comando `zfs receive`, nem a reversão nem o recebimento são concluídos e a seguinte mensagem de erro é devolvida:

```
cannot send 'pool/fs@name': not an earlier snapshot from the same fs
```

Enviando e recebendo fluxos complexos de instantâneos do ZFS

Esta seção descreve como usar as opções `zfs send -I` e `-R` para enviar e receber fluxos de instantâneos mais complexos.

Considere os pontos a seguir ao enviar e receber fluxos de instantâneos do ZFS complexos:

- Use a opção `zfs send -I` para enviar todos os fluxos incrementais de um instantâneo para um instantâneo cumulativo. Ou, utilize esta opção para enviar um fluxo incremental do instantâneo original para criar um clone. O instantâneo original já deve existir no lado de recepção para aceitar o fluxo incremental.
- Use a opção `zfs send -R` para enviar um fluxo de replicação de todos os sistemas de arquivos descendentes. Quando o fluxo de replicação é recebido, todas as propriedades, instantâneos, sistemas de arquivo descendentes e clones são preservados.

- Utilize ambas opções para enviar um fluxo de replicação incremental.
 - As alterações feitas às propriedades são preservadas, assim como as operações `rename` e `destroy` do instantâneo e do sistema de arquivos.
 - Se `zfs recv -F` não for especificado ao receber o fluxo de replicação, as operações `destroy` do conjunto de dados são ignoradas. A sintaxe de `zfs recv -F` neste caso também retém seu significado de *reversão se necessário*.
 - Como em outros casos (não `zfs send -R`) - `i` ou `-I`, se `-I` for usado, todos os instantâneos entre `snapA` e `snapD` são enviados. Se `-i` é utilizado, apenas `snapD` (para todos os descendentes) são enviados.
- Para receber qualquer um desses novos tipos de fluxos do `zfs send`, o sistema destinatário deve executar uma versão do software capaz de enviá-los. A versão de fluxo é incrementada. No entanto, você pode acessar os fluxos a partir das versões mais antigas do pool usando uma versão mais recente do software. Por exemplo, você pode enviar e receber os fluxos criados com as opções mais recentes para e de um pool de versão 3. Porém, você deve executar um software recente para receber um fluxo enviado com as opções mais recentes.

EXEMPLO 7-1 Enviando e recebendo fluxos complexos de instantâneos do ZFS

É possível combinar um grupo de instantâneos incrementais em um instantâneo usando a opção `zfs send -I`. Por exemplo:

```
# zfs send -I pool/fs@snapA pool/fs@snapD > /snaps/fs@all-I
```

Então, deve-se remover `snapB`, `snapC` e `snapD`.

```
# zfs destroy pool/fs@snapB
# zfs destroy pool/fs@snapC
# zfs destroy pool/fs@snapD
```

Para receber o instantâneo combinado, utilize o comando a seguir.

```
# 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	-

Você também pode usar o comando `zfs send -I` para combinar um instantâneo e um instantâneo de clone a fim de criar um conjunto de dados combinado. Por exemplo:

```
# zfs create pool/fs
# zfs snapshot pool/fs@snap1
# zfs clone pool/fs@snap1 pool/clone
# zfs snapshot pool/clone@snapA
```

EXEMPLO 7-1 Enviando e recebendo fluxos complexos de instantâneos do ZFS (Continuação)

```
# 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
```

Utilize o comando `zfs send -R` para replicar um sistema de arquivos ZFS e todos os sistemas de arquivos descendentes, até o instantâneo nomeado. Quando este fluxo é recebido, todas as propriedades, instantâneos, sistemas de arquivo descendentes e clones são preservados.

No exemplo seguinte, são criados instantâneos para os sistemas de arquivos do usuário. Um fluxo de replicação é criado para todos os instantâneos do usuário. A seguir, os instantâneos e sistemas de arquivos originais são destruídos e, então, recuperados.

```
# 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    -
```

No exemplo a seguir, o comando `zfs send -R` foi utilizado para replicar o conjunto de dados `users` e seus descendentes, e enviar o fluxo replicado para outro conjunto, `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    -
```

EXEMPLO 7-1 Enviando e recebendo fluxos complexos de instantâneos do ZFS *(Continuação)*

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	-

Replicação remota de dados do ZFS

Os comandos `zfs send` e `zfs recv` podem ser usados para copiar remotamente uma representação de fluxo do instantâneo de um sistema a outro. Por exemplo:

```
# zfs send tank/cindy@today | ssh newsys zfs recv sandbox/restfs@today
```

Este comando envia os dados do instantâneo `tank/cindy@today` e os recebe no sistema de arquivos `sandbox/restfs`. O comando também cria um instantâneo `restfs@today` no sistema `newsys`. Neste exemplo, o usuário foi configurado para que use `ssh` no sistema remoto.

Utilizando ACLs para proteger arquivos ZFS do Oracle Solaris

Este capítulo fornece informações sobre a utilização das listas de controle de acesso (ACLs) para proteger os arquivos do ZFS. As ACLs fornecem permissões mais granulares que as permissões do UNIX padrão.

Este capítulo traz as seguintes seções:

- “Novo modelo de ACL do Solaris” na página 237
- “Definindo ACLs em arquivos ZFS” na página 243
- “Definindo e exibindo ACLs em arquivos ZFS no formato verboso” na página 246
- “Definindo e exibindo ACLs em arquivos ZFS no formato compacto” na página 258

Novo modelo de ACL do Solaris

As versões anteriores do Solaris OS ofereciam suporte a uma implementação de ACL com base principalmente na especificação da ACL do rascunho POSIX. As ACLs baseadas no esquema POSIX são usadas para proteger arquivos UFS e são traduzidas por versões do NFS antes do NFSv4

Com a introdução do NFSv4, um novo modelo de ACL oferece suporte total à interoperabilidade que o NFSv4 oferece entre clientes e não-clientes UNIX. A nova implementação de ACL, conforme definida na especificação NFSv4, proporciona semânticas muito mais ricas baseadas nas ACLs de estilo NT.

As principais diferenças do novo modelo de ACL são as seguintes:

- O novo modelo de ACL tem base na especificação NFSv4 e é similar às ACLs de estilo NT.
- O novo modelo fornece um conjunto muito mais granular de privilégios de acesso. Para obter mais informações, consulte a [Tabela 8–2](#).
- ACLs são definidas e exibidas com os comandos `chmod` e `ls` em vez dos comandos `setfacl` e `getfacl`.

- O novo modelo oferece semânticas de herança mais ricas para designar como privilégios de acesso são aplicados de um diretório para subdiretórios, e assim por diante. Para obter mais informações, consulte [“Herança da ACL” na página 242](#).

Ambos os modelos de ACL proporcionam controles de acesso mais granulares que os disponíveis nas permissões de arquivos padrão. Assim como as ACLs de esquema POSIX, as novas ACLs compõem-se de várias entradas de controle de acesso (ACEs).

As ACLs com base no rascunho POSIX utilizam uma única entrada para definir quais permissões são aceitas e quais permissões são negadas. O novo modelo de ACL apresenta dois tipos de ACEs que afetam a verificação de acesso: ALLOW e DENY. Não é possível deduzir, a partir de qualquer ACE que defina um conjunto de permissões, se as permissões definidas nesta ACE são permitidas ou negadas.

A tradução entre ACLs do NFSv4 e de rascunho POSIX é a seguinte:

- Se utilizar utilitários com ACL, como os comandos `cp`, `mv`, `tar`, `cpio` ou `rcp` para transferir arquivos UFS com ACLs para um sistema de arquivos do ZFS, as ACLs de rascunho POSIX são traduzidas para as ACLs do NFSv4 equivalentes.
- Algumas ACLs do NFSv4 são traduzidas para ACLs de rascunho POSIX. Se uma ACL do NFSv4 não for traduzida para uma ACL de rascunho POSIX, será exibida a seguinte mensagem:

```
# cp -p filea /var/tmp
cp: failed to set acl entries on /var/tmp/filea
```

- Se criar um arquivo UFS `tar` ou `cpio` com a opção de preservar ACL (`tar -p` ou `cpio -P`) em um sistema que executa uma versão atual do Solaris, você perderá as ACLs quando o arquivo for extraído para um sistema que executa uma versão anterior do Solaris.

Todos os arquivos são extraídos com os modos de arquivo corretos, mas as entradas ACL são ignoradas.

- Você pode usar o comando `ufsrestore` para restaurar os dados dentro de um sistema de arquivos ZFS. Se os dados originais incluírem ACLs de rascunho POSIX, elas são traduzidas como ACLs NFSv4.
- Se tentar definir uma ACL do NFSv4 em um arquivo UFS, será exibida uma mensagem semelhante à seguinte:

```
chmod: ERROR: ACL type's are different
```

- Se tentar definir uma ACL de rascunho POSIX em um arquivo ZFS, será exibida uma mensagem semelhante à seguinte:

```
# getfacl filea
File system doesn't support aclent_t style ACL's.
See acl(5) for more information on Solaris ACL support.
```

Para obter mais informações sobre outras limitações com ACLs e produtos de backup, consulte [“Salvando dados do ZFS com outros produtos de backup” na página 231](#).

Descrições de sintaxe para definição de ACLs

Seguem dois formatos básicos de ACL:

Sintaxe para definição de ACLs comuns

Uma ACL é *comum* visto que representa somente as entradas *owner/group/other* tradicionais do UNIX.

```
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]- arquivo
```

Sintaxe para a configuração de ACLs incomuns

```
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]- arquivo
```

```
owner@, group@, everyone@
```

Identifica o *tipo de entrada ACL* da sintaxe da ACL comum. Para uma descrição dos tipos de entrada da ACL, consulte a [Tabela 8-1](#).

usuário ou *grupo*: *ACL-entry-ID=username or groupname*

Identifica o *tipo de entrada ACL* da sintaxe de ACL explícita. O usuário e grupo com *tipo de entrada ACL* devem conter também a *ID da entrada ACL*, *nome do usuário* ou *nome do grupo*. Para uma descrição dos tipos de entrada da ACL, consulte a [Tabela 8-1](#).

```
access-permissions/.../
```

Identifica as permissões de acesso que são aceitas ou negadas. Para obter uma descrição dos privilégios de acesso da ACL, consulte a [Tabela 8-2](#).

```
inheritance-flags
```

Identifica uma lista opcional de sinalizadores de herança da ACL. Para obter uma descrição dos sinalizadores de herança da ACL, consulte a [Tabela 8-3](#).

```
deny | allow
```

Identifica se as permissões de acesso são aceitas ou negadas.

No exemplo abaixo, o valor do *ID da entrada ACL* não é relevante:

```
group@:write_data/append_data/execute:deny
```

O exemplo abaixo inclui um *ID da entrada ACL* porque um usuário específico (*tipo da entrada ACL*) está incluído na ACL.

```
0:user:gozer:list_directory/read_data/execute:allow
```

Quando uma entrada ACL é exibida, ela se parece ao ilustrado abaixo:

```
2:group@:write_data/append_data/execute:deny
```

Neste exemplo, o **2** ou a designação da *index-ID* identifica a entrada ACL na ACL maior, que pode ter várias entradas para proprietário, UIDs específicos, grupo e todos. Você pode especificar o *ID de índice* com o comando `chmod` para identificar que parte da ACL quer modificar. É possível identificar, por exemplo, a ID de índice 3 como **A3** na sintaxe de comando `chmod`, semelhante ao seguinte:

```
chmod A3=user:venkman:read_acl:allow filename
```

Na tabela abaixo encontram-se as descrições dos tipos de entrada ACL, que são as representações da ACL de proprietário, grupo e outros.

TABELA 8-1 Tipos de entrada ACL

Tipo de entrada ACL	Descrição
owner@	Especifica o acesso concedido ao proprietário do objeto.
group@	Especifica o acesso concedido ao grupo proprietário do objeto.
everyone@	Especifica o acesso permitido a todos os usuários ou grupos que não correspondam com nenhuma outra entrada ACL.
user	Com um nome de usuário, especifica o acesso concedido a um usuário adicional do objeto. Esta entrada deve incluir a <i>ID da entrada ACL</i> , que contém um <i>nome de usuário</i> ou uma <i>ID de usuário</i> . Se o valor não for um UID numérico ou <i>nome de usuário</i> válidos, o tipo de entrada ACL é inválido.
group	Com um nome de grupo, especifica acesso concedido a um grupo adicional do objeto. Esta entrada deve incluir a <i>ID da entrada ACL</i> , que contém um <i>nome de grupo</i> ou uma <i>ID de grupo</i> . Se o valor não for um GID numérico ou <i>nome de grupo</i> válidos, o tipo de entrada ACL é inválido.

Os privilégios de acesso da ACL estão descritos na tabela abaixo.

TABELA 8-2 Privilégios de acesso da ACL

Privilégio de acesso	Privilégio de acesso compacto	Descrição
add_file	w	Permissão para adicionar um novo arquivo a um diretório.
add_subdirectory	p	Em um diretório, permissão para criar um subdiretório.

TABELA 8-2 Privilégios de acesso da ACL (Continuação)

Privilégio de acesso	Privilégio de acesso compacto	Descrição
append_data	p	Espaço reservado. Ainda não implementado.
delete	d	Permissão para excluir um arquivo.
delete_child	D	Permissão para excluir um arquivo ou um diretório dentro de um diretório.
execute	x	Permissão para executar um arquivo ou pesquisar o conteúdo de um diretório.
list_directory	a	Permissão para listar o conteúdo de um diretório.
read_acl	c	Permissão para ler a ACL (ls).
read_attributes	a	Permissão para ler os atributos básicos (não-ACLs) de um arquivo. Considere os atributos básicos como sendo os atributos do nível stat. Ao permitir este bit de máscara de acesso, a entidade pode executar ls(1) e stat(2).
read_data	a	Permissão para ler o conteúdo de um arquivo.
read_xattr	A	Permissão para ler os atributos estendidos de um arquivo ou efetuar uma consulta no diretório dos atributos estendidos do arquivo.
synchronize	s	Espaço reservado. Ainda não implementado.
write_xattr	W	Permissão para criar atributos estendidos ou gravar no diretório de atributos estendidos. A concessão de permissões a um usuário indica que o usuário pode criar um diretório de atributo estendido para um arquivo. As permissões do arquivo do atributo controlam o acesso do usuário ao atributo.
write_data	w	Permissão para modificar ou substituir o conteúdo de um arquivo.
write_attributes	m	Permissão para alterar os registros de data associados a um arquivo ou diretório para um valor arbitrário.
write_acl	C	Permissão para gravar a ACL ou modificar a ACL através do comando chmod.
write_owner	o	Permissão para alterar o proprietário ou o grupo do arquivo. Ou, a habilidade para executar os comandos chown ou chgrp no arquivo. Permissão para tomar a propriedade de um arquivo ou permissão para alterar a propriedade do grupo do arquivo para um grupo do qual o usuário é membro. Se quiser alterar a propriedade do grupo ou arquivo para um usuário ou grupo arbitrário, o privilégio PRIV_FILE_CHOWN é exigido.

Herança da ACL

O objetivo de utilizar a herança da ACL é que um arquivo ou diretório recém-criado pode herdar as ACLs que pretendem herdar, mas levando em consideração as permissões existentes no diretório pai.

Por padrão, as ACLs não são propagadas. Se definir uma ACL não-comum em um diretório, ela não será herdada por nenhum diretório subsequente. Você deve especificar a herança de uma ACL em um arquivo ou diretório.

Os sinalizadores de herança opcionais estão descritos na seguinte tabela.

TABELA 8-3 Sinalizadores de herança da ACL

Sinalizador de herança	Sinalizador de herança compacto	Descrição
<code>file_inherit</code>	<code>f</code>	Herda a ACL do diretório pai, mas se aplica apenas aos arquivos do diretório.
<code>dir_inherit</code>	<code>d</code>	Herda a ACL do diretório pai, mas se aplica apenas aos subdiretórios do diretório.
<code>inherit_only</code>	<code>i</code>	Herda a ACL do diretório pai, mas se aplica somente aos arquivos e subdiretórios recém-criados e não ao próprio diretório. Este sinalizador requer o sinalizador <code>file_inherit</code> , o sinalizador <code>dir_inherit</code> , ou ambos, para indicar o que herdar.
<code>no_propagate</code>	<code>n</code>	Herda somente a ACL do diretório pai até o conteúdo do primeiro nível do diretório, não herda o conteúdo do segundo nível ou dos níveis subsequentes. Este sinalizador requer o sinalizador <code>file_inherit</code> , o sinalizador <code>dir_inherit</code> , ou ambos, para indicar o que herdar.
-	N/D	Nenhuma permissão concedida.

Além disso, é possível definir a política de herança da ACL padrão no sistema de arquivos para que seja mais rigorosa ou menos rigorosa utilizando a propriedade `aclinherit` do sistema de arquivos. Para obter mais informações, consulte a próxima seção.

Propriedades da ACL

O sistema de arquivos do ZFS possui duas propriedades relacionados às ACLs:

- `aclinherit` – Esta propriedade determina o comportamento da herança da ACL. Os valores são os seguintes:

- `discard` – Para novos objetos, nenhuma entrada ACL é herdada quando um arquivo ou diretório é criado. A ACL no novo arquivo ou diretório é igual às permissões do arquivo ou diretório.
- `noallow` – Para novos objetos, somente as entradas ACLs herdáveis com um tipo de acesso `deny` são herdadas.
- `restricted` – Para novos objetos, as permissões `write_owner` e `write_acl` são removidas quando uma entrada ACL é herdada.
- `passthrough`: quando o valor da propriedade estiver definido como `passthrough`, os arquivos são criados com permissões determinadas pelas ACEs herdáveis. Se não existirem ACEs herdáveis que afetem as permissões, então as permissões serão definidas de acordo com as permissões solicitadas do aplicativo.
- `passthrough-x` : este valor de propriedade possui a mesma semântica de `passthrough`, exceto que, quando `passthrough-x` está ativado, os arquivos são criados com a permissão de execução (`x`), mas somente se a permissão de execução estiver definida no modo de criação de arquivo e em uma ACE de herança que afeta o modo.

O valor padrão da propriedade `aclinherit` é `restricted`.

- `aclmode`: esta propriedade modifica o comportamento da ACL quando um arquivo é inicialmente criado ou sempre que as permissões de um arquivo ou diretório forem modificadas pelo comando `chmod`. Os valores são os seguintes:
 - `discard` – Todas as entradas ACLs são removidas, exceto as entradas necessárias para definir o modo do arquivo ou diretório.
 - `groupmask`: as permissões ACL de usuário ou grupo são reduzidas para que não sejam maiores que as permissões de grupo, a menos que seja uma entrada de usuário com o mesmo UID que o proprietário do arquivo ou diretório. Assim, as permissões de ACL são reduzidas para que não sejam maiores que as permissões do proprietário.
 - `passthrough` – Durante uma operação `chmod`, as ACEs diferentes de `owner@`, `group@` ou `everyone@` não são modificadas de nenhuma forma. As ACEs com `owner@`, `group@` ou `everyone@` são desativadas para definir o modo de arquivo conforme solicitado pela operação `chmod`.

O valor padrão da propriedade `aclmode` é `groupmask`.

Definindo ACLs em arquivos ZFS

Por serem implementadas com o ZFS, as ACLs são compostas por entradas ACLs. O ZFS oferece um modelo de ACL *puro*, no qual todos os arquivos têm uma ACL. Normalmente, a ACL é *comum* porque representa somente entradas UNIX `owner/group/other` tradicionais.

Se alterar as permissões do arquivo, a ACL do arquivo será atualizada de acordo. Além disso, se remover uma ACL não-comum que dá ao usuário acesso ao arquivo ou diretório, tal usuário

poderia continuar tendo acesso ao arquivo ou diretório devido aos bits de permissão do arquivo ou diretório que permitem acesso para um grupo ou para todos. Todas as decisões de controles de acesso são controladas pelas permissões representadas em uma ACL do arquivo ou diretório.

As principais regras de acesso de ACL em um arquivo do ZFS são as seguintes:

- O ZFS processa as entradas ACLs com o objetivo de listá-las na ACL, de cima para baixo.
- São processadas somente as entradas ACLs que têm “quem” corresponda ao solicitante do acesso.
- Depois que a permissão `allow` foi concedida, ela não pode ser negada por uma entrada ACL `deny` subsequente no mesmo conjunto de permissões da ACL.
- A permissão `write_acl` é concedida incondicionalmente ao proprietário de um arquivo, mesmo se a permissão for explicitamente negada. Caso contrário, todas as permissões não especificadas são negadas.

No caso de permissões `deny`, ou quando falta um acesso de permissão, o subsistema de privilégios determina qual solicitação de acesso é concedida ao proprietário do arquivo ou ao superusuário. Este mecanismo evita que os proprietários de arquivos sejam bloqueados a seus arquivos e ativa o superusuário para que modifique os arquivos para recuperação.

Se definir uma ACL não-comum em um diretório, os filhos do diretório não herdarão a ACL automaticamente. Se definir uma ACL não-comum e quiser que ela seja herdada pelos filhos do diretório, você terá que utilizar os sinalizadores de herança da ACL. Para obter mais informações, consulte a [Tabela 8-3](#) e “[Definindo a herança da ACL em arquivos ZFS no formato verboso](#)” na [página 251](#).

Quando um novo arquivo é criado e dependendo do valor de `umask`, um padrão de ACL comum, semelhante ao ilustrado abaixo, é aplicado:

```
$ ls -v file.1
-rw-r--r--  1 root    root      206663 May 20 14:09 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
   /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
   /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
   :allow
```

Observe que cada categoria de usuário (`owner@`, `group@`, `everyone@`) possui duas entradas de ACL neste exemplo. Uma entrada para permissões `deny` e uma entrada para permissões `allow`.

A seguir, uma descrição dessa ACL do arquivo:

```
0:owner@      O proprietário não tem permissões de execução para o arquivo
               (execute:deny ).
```

- 1:owner@ O proprietário pode ler e modificar o conteúdo do arquivo (read_data/write_data/append_data). O proprietário também pode modificar os atributos do arquivo, tais como registros de data, atributos estendidos e ACLs (write_xattr/write_attributes /write_acl). Além disso, o proprietário pode modificar a propriedade do arquivo (write_owner:allow).
- 2:group@ O grupo não tem permissões para modificar e executar no arquivo (write_data/append_data/execute:deny).
- 3:group@ O grupo tem permissões de leitura para o arquivo (read_data:allow).
- 4:everyone@ Todos aqueles que não são proprietários de arquivos ou membros de um grupo de proprietários do arquivo não terão permissão para executar ou modificar o conteúdo de arquivos e nem para modificar os atributos do arquivo (write_data/append_data/write_xattr/execute/write_attributes /write_acl/write_owner:deny).
- 5:everyone@ Todos aqueles que não são proprietários de arquivos ou membros de um grupo de proprietários do arquivo possuem permissões no arquivo e em seus atributos (read_data/read_xattr/read_attributes/read_acl/synchronize:allow). A permissão de acesso synchronize não está implementada atualmente.

Quando um novo diretório é criado e dependendo do valor de umask, uma ACL de diretório padrão, semelhante à ilustrada a seguir, é aplicada:

```
$ ls -dv dir.1
drwxr-xr-x 2 root root 2 May 20 14:11 dir.1
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/write_xattr/execute/write_attributes/write_acl
/write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Segue uma descrição do diretório de ACL:

- 0:owner@ A lista deny do proprietário está vazia para o diretório (: : deny).
- 1:owner@ O proprietário pode ler e modificar o conteúdo do diretório (list_directory/read_data/add_file/write_data/add_subdirectory/append_data), pesquisa o conteúdo (execute) e modifica os atributos do arquivo, tais como registros de data, atributos

- estendidos e ACLs (`write_xattr/write_attributes/write_acl`). Além disso, o proprietário pode modificar a propriedade do diretório (`write_owner:allow`).
- 2:group@ O grupo não pode adicionar ou modificar o conteúdo do diretório (`add_file/write_data/add_subdirectory/append_data:deny`).
- 3:group@ O grupo pode listar e ler o conteúdo do diretório. Além disso, o grupo tem permissão para executar pesquisas no conteúdo do diretório (`list_directory/read_data/execute:allow`).
- 4:everyone@ Todos aqueles que não são proprietários de arquivos ou membros de um grupo de proprietários do arquivo não terão permissão para adicionar ou modificar o conteúdo do diretório (`add_file/write_data/add_subdirectory/append_data`). Além disso, a permissão para modificar qualquer um dos atributos do diretório é negada (`write_xattr/write_attributes/write_acl/write_owner:deny`).
- 5:everyone@ Todos aqueles que não são proprietários de arquivos ou membros de um grupo de proprietários do arquivo terão permissão de leitura e execução no conteúdo e nos atributos do diretório (`list_directory/read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow`). A permissão de acesso `synchronize` não está implementada atualmente.

Definindo e exibindo ACLs em arquivos ZFS no formato verboso

O comando `chmod` pode ser usado para modificar as ACLs em arquivos ZFS. A sintaxe de `chmod` abaixo para a modificação de ACLs usa *acl-specification* para identificar o formato da ACL. Para obter uma descrição de *acl-specification*, consulte “[Descrições de sintaxe para definição de ACLs](#)” na página 239.

- Adicionando entradas ACL
 - Adicionando uma entrada ACL de um usuário
 - % `chmod A+acl-specification filename`
 - Adicionando uma entrada ACL por *ID de índice*
 - % `chmod Aindex-ID+acl-specification filename`

Esta sintaxe insere a nova entrada ACL no local do *ID de índice* especificado.

- Substituindo uma entrada ACL

- % `chmod A=acl-specification filename`
- % `chmod Aindex-ID=acl-specification filename`
- Removendo entradas ACLs
 - Removendo uma entrada ACL por *ID de índice*
 - % `chmod Aindex-ID- filename`
 - Removendo uma entrada ACL por usuário
 - % `chmod A-acl-specification filename`
 - Removendo todas as ACEs não-comuns de um arquivo
 - % `chmod A- filename`

As informações detalhadas da ACL são exibidas com o comando `ls -v`. Por exemplo:

```
# ls -v file.1
-rw-r--r-- 1 root    root      206663 May 20 14:09 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
    /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
    :allow
```

Para obter mais informações sobre o uso do formato compacto da ACL, consulte [“Definindo e exibindo ACLs em arquivos ZFS no formato compacto” na página 258](#).

EXEMPLO 8-1 Modificando ACLs comuns em arquivos ZFS

Esta seção oferece exemplos de configuração e exibição de ACLs comuns.

No exemplo abaixo, existe uma ACL comum no `file.1`:

```
# ls -v file.1
-rw-r--r-- 1 root    root      206663 May 20 15:03 file.1
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
    /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
    :allow
```

No exemplo abaixo, as permissões `write_data` são concedidas a `group@`:

```
# chmod A2=group@:append_data/execute:deny file.1
# chmod A3=group@:read_data/write_data:allow file.1
```

EXEMPLO 8-1 Modificando ACLs comuns em arquivos ZFS (Continuação)

```
# ls -v file.1
-rw-rw-r-- 1 root    root      206663 May 20 15:03 file.1
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:append_data/execute:deny
3:group@:read_data/write_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

No exemplo abaixo, as permissões no file.1 são redefinidas como 644.

```
# chmod 644 file.1
# ls -v file.1
-rw-r--r-- 1 root    root      206663 May 20 15:03 file.1
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

EXEMPLO 8-2 Definindo ACLs não-comuns em arquivos ZFS

Esta seção oferece exemplos de definições e exibições de ACLs não-comuns.

No exemplo a seguir, as permissões read_data/execute são adicionadas ao usuário gozer no diretório test.dir:

```
# chmod A+user:gozer:read_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+ 2 root    root      2 May 20 15:09 test.dir
0:user:gozer:list_directory/read_data/execute:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

No exemplo abaixo, as permissões read_data/execute são removidas do usuário gozer:

EXEMPLO 8-2 Definindo ACLs não-comuns em arquivos ZFS (Continuação)

```
# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x  2 root      root          2 May 20 15:09 test.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

EXEMPLO 8-3 A interação da ACL com permissões em arquivos ZFS

Estes exemplos ilustram a interação entre definir ACLs e alterar as permissões do arquivo ou diretório.

No exemplo abaixo, existe uma ACL comum em `file.2`:

```
# ls -v file.2
-rw-r--r--  1 root      root          3103 May 20 15:23 file.2
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

No exemplo a seguir, as permissões `allow` da ACL são removidas de `everyone@`:

```
# chmod A5- file.2
# ls -v file.2
-rw-r-----+ 1 root      root          3103 May 20 15:23 file.2
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
```

Nesta saída, as permissões do arquivo são redefinidos de 644 para 640. As permissões de leitura para `everyone@` foram removidas com sucesso das permissões do arquivo quando as permissões `allow` da ACL foram removidas de `everyone@`.

EXEMPLO 8-3 A interação da ACL com permissões em arquivos ZFS (Continuação)

No exemplo abaixo, a ACL existente é substituída pelas permissões `read_data/write_data` em `everyone@`:

```
# chmod A=everyone@:read_data/write_data:allow file.3
# ls -v file.3
-rw-rw-rw-+ 1 root    root        6986 May 20 15:25 file.3
    0:everyone@:read_data/write_data:allow
```

Nessa saída, a sintaxe `chmod` substitui com sucesso a ACL existente pelas permissões `read_data/write_data:allow` para permitir leitura/gravação em `owner`, `group` e `everyone@`. Nesse modelo, `everyone@` especifica para qualquer usuário ou grupo. Visto que não existe nenhuma entrada ACL de `owner@` ou de `group@` para substituir as permissões para proprietário e grupo, as permissões são definidas como `666`.

No exemplo abaixo, a ACL existente é substituída pelas permissões de leitura no usuário `gozer`:

```
# chmod A=user:gozer:read_data:allow file.3
# ls -v file.3
-----+ 1 root    root        6986 May 20 15:25 file.3
    0:user:gozer:read_data:allow
```

Nessa saída, as permissões do arquivo são calculadas para que sejam `000` porque não existe nenhuma entrada ACL em `owner@`, `group@` ou `everyone@`, que representam os componentes de permissões tradicionais de um arquivo. O proprietário do arquivo pode solucionar esse problema redefinindo as permissões (e a ACL) da seguinte forma:

```
# chmod 655 file.3
# ls -v file.3
-rw-r-xr-x+ 1 root    root        6986 May 20 15:25 file.3
    0:user:gozer::deny
    1:user:gozer:read_data:allow
    2:owner@:execute:deny
    3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
      /write_acl/write_owner:allow
    4:group@:write_data/append_data:deny
    5:group@:read_data/execute:allow
    6:everyone@:write_data/append_data/write_xattr/write_attributes
      /write_acl/write_owner:deny
    7:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
      /synchronize:allow
```

EXEMPLO 8-4 Restaurando ACLs comuns em arquivos ZFS

É possível utilizar o comando `chmod` para remover todas as ACLs não-comuns em um arquivo ou diretório e, assim, restaurar as ACLs comuns no arquivo ou diretório.

No exemplo abaixo, existem duas ACEs não-comuns em `test5.dir`:

EXEMPLO 8-4 Restaurando ACLs comuns em arquivos ZFS (Continuação)

```
# ls -dv test5.dir
drwxr-xr-x  2 root    root          2 May 20 15:32 test5.dir
0:user:lp:read_data:file_inherit:deny
1:user:gozer:read_data:file_inherit:deny
2:owner@::deny
3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
4:group@:add_file/write_data/add_subdirectory/append_data:deny
5:group@:list_directory/read_data/execute:allow
6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
7:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

No exemplo abaixo, as ACLs não-comuns de usuários gozer e lp são removidas. A ACL restante contém seis valores padrão em owner@, group@ e everyone@.

```
# chmod A- test5.dir
# ls -dv test5.dir
drwxr-xr-x  2 root    root          2 May 20 15:32 test5.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Definindo a herança da ACL em arquivos ZFS no formato verboso

É possível especificar se e como as ACLs são herdadas nos arquivos do diretório. Por padrão, as ACLs não são propagadas. Se uma ACL não-comum for definida em um diretório, a ACL não será herdada por nenhum diretório subsequente. É possível especificar a herança de uma ACL em um arquivo ou diretório.

Além disso, são oferecidas duas propriedades de ACL que podem ser definidas globalmente em sistemas de arquivos: `aclinherit` e `aclmode`. Por padrão, `aclinherit` é definida como `restricted` e `aclmode` é definida como `groupmask`.

Para obter mais informações, consulte [“Herança da ACL” na página 242](#).

EXEMPLO 8-5 Concedendo herança de ACL padrão

Por padrão, as ACLs não se propagam pela estrutura do diretório.

No exemplo seguinte, uma ACE não-comum de `read_data/write_data/execute` é aplicada para o usuário `gozer` no diretório `test.dir`:

```
# chmod A+user:gozer:read_data/write_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+ 2 root      root          2 May 20 15:41 test.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Se for criado um subdiretório `test.dir`, a ACE não se propaga para o usuário `gozer`. O usuário `gozer` teria acesso ao subdiretório somente se as permissões no subdiretório lhe permitissem acessar como proprietário de arquivo, membro de grupo ou `everyone@`. Por exemplo:

```
# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x 2 root      root          2 May 20 15:42 test.dir/sub.dir
 0:owner@::deny
 1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 2:group@:add_file/write_data/add_subdirectory/append_data:deny
 3:group@:list_directory/read_data/execute:allow
 4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

EXEMPLO 8-6 Permitindo herança de ACL em arquivos e diretórios

Os exemplos a seguir identificam as ACEs do arquivo e do diretório que são aplicadas quando o sinalizador `file_inherit` está definido.

Neste exemplo, as permissões `read_data/write_data` são adicionadas para arquivos do diretório `test.dir` do usuário `gozer`, para que ele tenha acesso de leitura a quaisquer arquivos recém-criados:

```
# chmod A+user:gozer:read_data/write_data:file_inherit:allow test2.dir
# ls -dv test2.dir
drwxr-xr-x+ 2 root      root          2 May 20 15:50 test2.dir
```

EXEMPLO 8-6 Permitindo herança de ACL em arquivos e diretórios *(Continuação)*

```

0:user:gozer:read_data/write_data:file_inherit:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

Neste exemplo, as permissões do usuário gozer são aplicadas no arquivo `test2.dir/file.2` recém-criado. A herança de ACL fornecida, `read_data:file_inherit:allow`, indica que o usuário gozer pode ler o conteúdo de todos os arquivos recém-criados.

```

# touch test2.dir/file.2
# ls -lv test2.dir/file.2
-rw-r--r--+ 1 root   root           0 May 20 15:51 test2.dir/file.2
  0:user:gozer:write_data:deny
  1:user:gozer:read_data/write_data:allow
  2:owner@:execute:deny
  3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:allow
  4:group@:write_data/append_data/execute:deny
  5:group@:read_data:allow
  6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
    /write_acl/write_owner:deny
  7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
    :allow

```

Pelo fato da propriedade `aclmode` para este arquivo estar definida com o valor padrão (`groupmask`), o usuário gozer não tem a permissão `write_data` no `file.2` porque a permissão de grupo do arquivo não lhe permite.

Ao utilizar a permissão `inherit_only`, aplicada quando os sinalizadores `file_inherit` ou `dir_inherit` são definidos, é utilizada para propagar a ACL pela estrutura do diretório. Como tal, o usuário gozer tem permissão concedida ou negada apenas a partir das permissões de `everyone@`, a não ser que ele seja o proprietário do arquivo ou um membro do grupo ao qual o arquivo pertence. Por exemplo:

```

# mkdir test2.dir/subdir.2
# ls -dv test2.dir/subdir.2
drwxr-xr-x+ 2 root   root           2 May 20 15:52 test2.dir/subdir.2
  0:user:gozer:list_directory/read_data/add_file/write_data:file_inherit
    /inherit_only:allow
  1:owner@::deny
  2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
    /append_data/write_xattr/execute/write_attributes/write_acl
    /write_owner:allow
  3:group@:add_file/write_data/add_subdirectory/append_data:deny

```

EXEMPLO 8-6 Permitindo herança de ACL em arquivos e diretórios *(Continuação)*

```

4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

As séries de exemplos a seguir identificam as ACLs do arquivo e do diretório que são aplicadas quando ambos sinalizadores, `file_inherit` e `dir_inherit`, estão definidos.

Neste exemplo, o usuário `gozer` tem permissões de leitura, gravação e execução herdadas por arquivos e diretórios recém-criados:

```

# chmod A+user:gozer:read_data/write_data/execute:file_inherit/dir_inherit:allow
test3.dir
# ls -dv test3.dir
drwxr-xr-x+ 2 root    root          2 May 20 15:53 test3.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
  :file_inherit/dir_inherit:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

# touch test3.dir/file.3
# ls -v test3.dir/file.3
-rw-r--r--+ 1 root    root          0 May 20 15:58 test3.dir/file.3
 0:user:gozer:write_data/execute:deny
 1:user:gozer:read_data/write_data/execute:allow
 2:owner@:execute:deny
 3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
 4:group@:write_data/append_data/execute:deny
 5:group@:read_data:allow
 6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
 7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow

# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+ 2 root    root          2 May 20 15:59 test3.dir/subdir.1
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
  :file_inherit/dir_inherit/inherit_only:allow
 1:user:gozer:add_file/write_data:deny
 2:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
 3:owner@::deny
 4:owner@:list_directory/read_data/add_file/write_data/add_subdirectory

```

EXEMPLO 8-6 Permitindo herança de ACL em arquivos e diretórios *(Continuação)*

```

    /append_data/write_xattr/execute/write_attributes/write_acl
    /write_owner:allow
5:group@:add_file/write_data/add_subdirectory/append_data:deny
6:group@:list_directory/read_data/execute:allow
7:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
8:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

Nestes exemplos, pelo fato de as permissões do diretório pai para `group@` e `everyone@` negarem as permissões de gravação e execução, o usuário `gozer` não tem permissões de gravação e execução. A propriedade `aclmode` padrão é `restricted`, o que indica que as permissões `write_data` e `execute` não são herdadas.

Neste exemplo, o usuário `gozer` tem permissões de leitura, gravação e execução herdadas por arquivos recém-criados. Entretanto, as ACLs não são propagadas para conteúdos subsequentes do diretório.

```

# chmod A+user:gozer:read_data/write_data/execute:file_inherit/no_propagate:allow
test4.dir
# ls -dv test4.dir
drwxr-xr-x  2 root      root           2 May 20 16:02 test4.dir
0:user:gozer:list_directory/read_data/add_file/write_data/execute
  :file_inherit/no_propagate:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

Como mostra o exemplo a seguir, quando um novo subdiretório é criado, a permissão para arquivos `read_data/write_data/execute` do usuário `gozer` não se propaga para o novo diretório `sub4.dir`:

```

mkdir test4.dir/sub4.dir
# ls -dv test4.dir/sub4.dir
drwxr-xr-x  2 root      root           2 May 20 16:03 test4.dir/sub4.dir
0:owner@::deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
2:group@:add_file/write_data/add_subdirectory/append_data:deny
3:group@:list_directory/read_data/execute:allow
4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny

```

EXEMPLO 8-6 Permitindo herança de ACL em arquivos e diretórios *(Continuação)*

```
5:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

Conforme ilustra o exemplo a seguir, as permissões `read_data/write_data/execute` do usuário `gozer` se propagam para o arquivo recém-criado:

```
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 May 20 16:04 test4.dir/file.4
0:user:gozer:write_data/execute:deny
1:user:gozer:read_data/write_data/execute:allow
2:owner@:execute:deny
3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
/write_acl/write_owner:allow
4:group@:write_data/append_data/execute:deny
5:group@:read_data:allow
6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
/write_acl/write_owner:deny
7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

EXEMPLO 8-7 Herança de ACL com a propriedade `aclmode` definida como `passthrough`

Como mostra o exemplo a seguir, quando a propriedade `aclmode` no sistema de arquivos `tank/cindys` é definida como `passthrough`, o usuário `gozer` herda as ACLs aplicadas ao diretório `test4.dir` para o arquivo `file.4` recém-criado:

```
# zfs set aclmode=passthrough tank/cindys
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 May 20 16:08 test4.dir/file.4
0:user:gozer:write_data/execute:deny
1:user:gozer:read_data/write_data/execute:allow
2:owner@:execute:deny
3:owner@:read_data/write_data/append_data/write_xattr/write_attributes
/write_acl/write_owner:allow
4:group@:write_data/append_data/execute:deny
5:group@:read_data:allow
6:everyone@:write_data/append_data/write_xattr/execute/write_attributes
/write_acl/write_owner:deny
7:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Essa saída mostra que a ACL

`read_data/write_data/execute:allow:file_inherit/dir_inherit` que foi definida no diretório pai `test4.dir` é passada para o usuário `gozer`.

EXEMPLO 8-8 Herança de ACL com a propriedade `aclmode` definida como `discard`

Se a propriedade `aclmode` estiver definida como `discard` em um sistema de arquivos, as ACLs podem ser potencialmente descartadas quando as permissões de um diretório forem alteradas. Por exemplo:

```
# zfs set aclmode=discard tank/cindys
# chmod A+user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
# ls -dv test5.dir
drwxr-xr-x+ 2 root    root          2 May 20 16:09 test5.dir
 0:user:gozer:list_directory/read_data/add_file/write_data/execute
   :dir_inherit:allow
 1:owner@::deny
 2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
   /append_data/write_xattr/execute/write_attributes/write_acl
   /write_owner:allow
 3:group@:add_file/write_data/add_subdirectory/append_data:deny
 4:group@:list_directory/read_data/execute:allow
 5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
   /write_attributes/write_acl/write_owner:deny
 6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
   /read_acl/synchronize:allow
```

Se, mais tarde, você decidir comprimir as permissões em um diretório, a ACL não-comum é descartada. Por exemplo:

```
# chmod 744 test5.dir
# ls -dv test5.dir
drwxr--r-- 2 root    root          2 May 20 16:09 test5.dir
 0:owner@::deny
 1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
   /append_data/write_xattr/execute/write_attributes/write_acl
   /write_owner:allow
 2:group@:add_file/write_data/add_subdirectory/append_data/execute:deny
 3:group@:list_directory/read_data:allow
 4:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
   /execute/write_attributes/write_acl/write_owner:deny
 5:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
   /synchronize:allow
```

EXEMPLO 8-9 Herança de ACL com a propriedade `aclinherit` definida como `noallow`

No exemplo abaixo, estão definidas duas ACLs não-comuns com herança de arquivo. Uma ACL concede a permissão `read_data` e outra ACL nega a permissão `read_data`. Este exemplo também mostra como você pode especificar duas ACEs no mesmo comando `chmod`.

```
# zfs set aclinherit=noallow tank/cindys
# chmod A+user:gozer:read_data:file_inherit:deny,user:lp:read_data:file_inherit:allow
test6.dir
# ls -dv test6.dir
drwxr-xr-x+ 2 root    root          2 May 20 16:11 test6.dir
 0:user:gozer:read_data:file_inherit:deny
 1:user:lp:read_data:file_inherit:allow
 2:owner@::deny
```

EXEMPLO 8-9 Herança de ACL com a propriedade `aclinherit` definida como `noallow` (Continuação)

```
3:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
4:group@:add_file/write_data/add_subdirectory/append_data:deny
5:group@:list_directory/read_data/execute:allow
6:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
  /write_attributes/write_acl/write_owner:deny
7:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Conforme mostra o exemplo abaixo, quando um novo arquivo é criado, a ACL que concede a permissão `read_data` é descartada.

```
# touch test6.dir/file.6
# ls -v test6.dir/file.6
-rw-r--r--  1 root    root          0 May 20 16:13 test6.dir/file.6
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

Definindo e exibindo ACLs em arquivos ZFS no formato compacto

Você pode definir e exibir as permissões em arquivos ZFS em um formato compacto que usa unicamente 14 letras para representar as permissões. As letras que representam as permissões compactas estão listadas na [Tabela 8-2](#) e [Tabela 8-3](#).

Você pode exibir listagens de ACLs compactas de arquivos e diretórios usando o comando `ls -V`. Por exemplo:

```
# ls -V file.1
-rw-r--r--  1 root    root          206663 Jun 17 10:07 file.1
  owner@: -x-----:-----:deny
  owner@: rw-p---A-W-Co-:-----:allow
  group@: -wxp-----:-----:deny
  group@: r-----:-----:allow
  everyone@: -wxp---A-W-Co-:-----:deny
  everyone@: r-----a-R-c-s:-----:allow
```

A saída de ACL compacta é descrita da seguinte forma:

`owner@` O proprietário não tem permissões de execução para o arquivo (x= execute).

owner@	O proprietário pode ler e modificar o conteúdo do arquivo (rw=read_data/write_data), (p= append_data). O proprietário também pode modificar os atributos do arquivo, tais como registros de data, atributos estendidos e ACLs (A=write_xattr, W=write_attributes e C=write_acl). Além disso, o proprietário pode modificar a propriedade do arquivo (o=write_owner).
group@	O grupo não tem permissões de modificação e execução para o arquivo (write_data, p=append_data e x=execute).
group@	O grupo tem permissões de leitura para o arquivo (r= read_data).
everyone@	Todos os que não são proprietários do arquivo ou membros do grupo de proprietários do arquivo não têm permissão para executar ou modificar o conteúdo do arquivo nem para modificar os atributos do arquivo (w=write_data, x=execute, p =append_data, A=write_xattr, W=write_attributes, C= write_acl e o=write_owner).
everyone@	Todos aqueles que não são proprietários de arquivos ou membros de um grupo de proprietários do arquivo em um arquivo e em seus atributos (r= read_data, a=append_data, R=read_xattr, c=read_acl e s=synchronize). A permissão de acesso synchronize não está implementada atualmente.

A ACL de formato compacto oferece as seguintes vantagens em relação à ACL de formato verboso:

- As permissões podem ser especificadas como argumentos posicionais para o comando `chmod`.
- O caractere hífen (-), que identifica a ausência de permissões, pode ser removido. Apenas as letras requisitadas precisam ser especificadas.
- Ambos sinalizadores, de permissões e de herança, estão definidos no mesmo modo.

Para obter mais informações sobre o uso de ACL de formato compacto, consulte [“Definindo e exibindo ACLs em arquivos ZFS no formato verboso” na página 246](#).

EXEMPLO 8-10 Definindo e exibindo ACLs em formato compacto

No exemplo a seguir, existe uma ACL comum no `file.1`:

```
# ls -V file.1
-rw-r--r-- 1 root    root      206663 Jun 17 10:07 file.1
owner@:--x-----:-----:deny
owner@:rw-p--A-W-Co-:-----:allow
group@:-wxp-----:-----:deny
group@:r-----:-----:allow
everyone@:-wxp--A-W-Co-:-----:deny
everyone@:r-----a-R-c--s:-----:allow
```

EXEMPLO 8-10 Definindo e exibindo ACLs em formato compacto (Continuação)

No exemplo seguinte, as permissões `read_data/execute` são adicionadas ao usuário `gozer` no diretório `file.1`:

```
# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+ 1 root    root        206663 Jun 17 10:07 file.1
  user:gozer:r-x-----:-----:allow
  owner@:--x-----:-----:deny
  owner@:rw-p--A-W-Co-:-----:allow
  group@:-wxp-----:-----:deny
  group@:r-----:-----:allow
  everyone@:-wxp--A-W-Co-:-----:deny
  everyone@:r-----a-R-c--s:-----:allow
```

Outra forma de adicionar as mesmas permissões para o usuário `gozer` é inserir uma nova entrada ACL em uma posição específica, 4, por exemplo. As ACLs existentes nas posições 4–6 são levadas para baixo. Por exemplo:

```
# chmod A4+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+ 1 root    root        206663 Jun 17 10:16 file.1
  owner@:--x-----:-----:deny
  owner@:rw-p--A-W-Co-:-----:allow
  group@:-wxp-----:-----:deny
  group@:r-----:-----:allow
  user:gozer:r-x-----:-----:allow
  everyone@:-wxp--A-W-Co-:-----:deny
  everyone@:r-----a-R-c--s:-----:allow
```

No exemplo abaixo, o usuário `gozer` tem permissões de leitura, gravação e execução herdadas por arquivos e diretórios recém-criados.

```
# chmod A+user:gozer:rwx:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+ 2 root    root          2 Jun 17 10:19 dir.2
  user:gozer:rwx-----:fd----:allow
  owner@:-----:-----:deny
  owner@:rwxp--A-W-Co-:-----:allow
  group@:-w-p-----:-----:deny
  group@:r-x-----:-----:allow
  everyone@:-w-p--A-W-Co-:-----:deny
  everyone@:r-x---a-R-c--s:-----:allow
```

Você também pode recortar e colar os sinalizadores de permissões e de herança da saída `ls -V` no formato compacto `chmod`. Por exemplo, para duplicar os sinalizadores de permissões e de herança no diretório `dir.2` do usuário `gozer` para o usuário `cindys` em `dir.2`, copie e cole os sinalizadores de permissão e de herança (`rwx-----:f-----:allow`) no comando `chmod`, como mostra a seguir:

EXEMPLO 8-10 Definindo e exibindo ACLs em formato compacto (Continuação)

```
# chmod A+user:cindys:rwX-----:fd----:allow dir.2
# ls -dV dir.2
drwxr-xr-x+ 2 root    root          2 Jun 17 10:19 dir.2
      user:cindys:rwX-----:fd----:allow
      user:gozer:rwX-----:fd----:allow
      owner@:-----:-----:deny
      owner@:rwxp---A-W-Co:-----:allow
      group@:-w-p-----:-----:deny
      group@:r-x-----:-----:allow
      everyone@:-w-p---A-W-Co:-----:deny
      everyone@:r-x---a-R-C--s:-----:allow
```

EXEMPLO 8-11 Herança de ACL com a propriedade `aclinherit` definida como `passthrough`

Um sistema de arquivos que apresenta a propriedade `aclinherit` definida como `passthrough` herda todas as entradas da ACL sem as modificações feitas em tais entradas quando estas são herdadas. Quando esta propriedade estiver definida como `passthrough`, os arquivos serão criados com permissões determinadas pelas ACEs herdáveis. Se não existirem ACEs herdáveis que afetem as permissões, então as permissões serão definidas de acordo com as permissões solicitadas do aplicativo.

Os exemplos seguintes utilizam a sintaxe compacta de ACL para mostrar como herdar permissões definindo a propriedade `aclinherit` como `passthrough`.

Neste exemplo, uma ACL está definida no diretório `test1.dir` para forçar a herança. A sintaxe cria uma entrada ACL de `owner@`, `group@` e `everyone@` para os arquivos recém-criados. O diretórios recém-criados herdam uma entrada ACL de `@owner`, `group@` e `everyone@`. Adicionalmente, os diretórios herdam outras seis ACEs que propagam as ACEs para os arquivos e diretórios recém-criados.

```
# zfs set aclinherit=passthrough tank/cindys
# pwd
/tank/cindys
# mkdir test1.dir

# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow
test1.dir
# ls -Vd test1.dir
drwxrwx---+ 2 root    root          2 Jun 17 10:37 test1.dir
      owner@:rwxpdDaARwCcos:fd----:allow
      group@:rwxp-----:fd----:allow
      everyone@:-----:fd----:allow
```

Neste exemplo, um arquivo recém-criado herda a ACL que foi especificada para ser herdada pelos arquivos recém-criados.

```
# cd test1.dir
# touch file.1
# ls -V file.1
```

EXEMPLO 8-11 Herança de ACL com a propriedade `aclinherit` definida como `passthrough` (Continuação)

```
-rwxrwx---+ 1 root    root          0 Jun 17 10:38 file.1
      owner@: rwxpdDaARwCcos:-----:allow
      group@: rwxp-----:-----:allow
      everyone@:-----:-----:allow
```

Neste exemplo, o diretório recém-criado herda as ACEs que controlam o acesso a este diretório, bem como as ACEs para futuras propagações para os filhos do diretório recém-criado.

```
# mkdir subdir.1
# ls -dV subdir.1
drwxrwx---+ 2 root    root          2 Jun 17 10:39 subdir.1
      owner@: rwxpdDaARwCcos:fdi---:allow
      owner@: rwxpdDaARwCcos:-----:allow
      group@: rwxp-----:fdi---:allow
      group@: rwxp-----:-----:allow
      everyone@:-----:fdi---:allow
      everyone@:-----:-----:allow
```

As entradas `-di--` e `f-i---` servem para propagar herança e não são consideradas durante o controle de acesso. Neste exemplo, é criado um arquivo com uma ACL comum em outro diretório no qual as ACEs herdadas não estão presentes.

```
# cd /tank/cindys
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r-- 1 root    root          0 Jun 17 10:40 file.2
      owner@: -x-----:-----:deny
      owner@: rw-p--A-W-Co-:-----:allow
      group@: -wpx-----:-----:deny
      group@: r-----:-----:allow
      everyone@: -wpx--A-W-Co-:-----:deny
      everyone@: r-----a-R-c-s:-----:allow
```

EXEMPLO 8-12 Herança de ACL com a propriedade `aclinherit` definida como `passthrough-x`

Quando a propriedade `aclinherit` está definida como `passthrough-x`, os arquivos são criados com permissão `x` para executar para `owner@`, `group@` ou `everyone@`, mas somente se a permissão de executar estiver definida no modo de criação de arquivo em uma ACE herdável que afeta o modo.

Os exemplos a seguir mostram como herdar a permissão de executar ao definir a propriedade `aclinherit` como `passthrough-x`.

```
# zfs set aclinherit=passthrough-x tank/cindys
```

A ACL a seguir é definida em `/tank/cindys/test1.dir` para fornecer uma herança ACL executável para arquivos para `owner@`, `group@` e `everyone@`.

EXEMPLO 8-12 Herança de ACL com a propriedade `aclinherit` definida como `passthrough-x`
(*Continuação*)

```
# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow test1.dir
# ls -Vd test1.dir
drwxrwx---+ 2 root      root          2 Jun 17 10:41 test1.dir
      owner@:rwxpdDaARWcCos:fd----:allow
      group@:rwxp-----:fd----:allow
      everyone@:-----:fd----:allow
```

Um arquivo (`file1`) é criado com as permissões solicitadas `0666`. As permissões resultantes são `0660`. A permissão de execução não foi herdada já que o modo de criação não a solicitou.

```
# touch test1.dir/file1
# ls -V test1.dir/file1
-rw-rw---+ 1 root      root          0 Jun 17 10:42 test1.dir/file1
      owner@:rw-pdDaARWcCos:-----:allow
      group@:rw-p-----:-----:allow
      everyone@:-----:-----:allow
```

A seguir, um executável denominado `t` é gerado com o uso do compilador `cc` no diretório `testdir`.

```
# cc -o t t.c
# ls -V t
-rwxrwx---+ 1 root      root          7396 Jun 17 10:50 t
      owner@:rwxpdDaARWcCos:-----:allow
      group@:rwxp-----:-----:allow
      everyone@:-----:-----:allow
```

As permissões resultantes são `0770` porque o `cc` solicitou permissões `0777`, que causou a herança da permissão de executar das entradas `owner@`, `group@` e `everyone@`.

Administração delegada de ZFS

Este capítulo descreve como usar a administração delegada para permitir que usuários não-privilegiados realizem tarefas de administração no ZFS.

- “Visão geral da administração delegada de ZFS” na página 265
- “Delegando permissões do ZFS” na página 266
- “Exibindo permissões delegadas do ZFS (Exemplos)” na página 274
- “Delegando permissões do ZFS (Exemplos)” na página 270
- “Removendo permissões do ZFS (Exemplos)” na página 275

Visão geral da administração delegada de ZFS

Este recurso permite distribuir permissões refinadas a usuários específicos, grupos ou a todos. São suportados dois tipos de permissões delegadas:

- As permissões individuais podem ser explicitamente especificadas como criar, destruir, montar, realizar um instantâneo, etc.
- Os grupos de permissões denominados *conjuntos de permissões* podem ser definidos. Um conjunto de permissões pode ser atualizado mais tarde e todos os consumidores deste conjunto adquirem automaticamente a alteração. Os conjuntos de permissões começam com o caractere @ e estão limitados a 64 caracteres de extensão. Depois do caractere @, os caracteres restantes do nome do conjunto apresentam as mesmas restrições que os nomes de um sistema de arquivos ZFS normal.

A administração delegada de ZFS oferece recursos semelhantes ao modelo de segurança RBAC. Este recurso oferece as seguintes vantagens na administração de conjuntos de armazenamento e sistemas de arquivos do ZFS:

- As permissões seguem o pool de armazenamento do ZFS quando tal pool for migrado.
- Oferece herança dinâmica, onde é possível controlar como as permissões se propagam pelos sistemas de arquivos.

- Pode ser configurado de forma que somente o criador de um sistema de arquivos possa destruir tal sistema de arquivos.
- Você pode distribuir permissões para especificar sistemas de arquivos. Os sistemas de arquivos recém-criados podem adquirir automaticamente as permissões.
- Fornece administração do NFS simples. Por exemplo, um usuário com permissões explícitas pode criar um instantâneo sobre o NFS no diretório `.zfs/snapshot` apropriado.

Considere o uso da administração delegada para distribuir tarefas do ZFS. Para obter mais informações sobre o uso de RBAC para gerenciar tarefas gerais de administração do Solaris, consulte [Parte III, “Roles, Rights Profiles, and Privileges,” no *System Administration Guide: Security Services*](#).

Desativando permissões delegadas do ZFS

É possível controlar os recursos de administração delegados utilizando a propriedade `delegation` do conjunto. Por exemplo:

```
# 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
```

Por padrão, a propriedade `delegation` é habilitada.

Delegando permissões do ZFS

Você pode usar o comando `zfs allow` a fim de conceder permissões em conjuntos de dados ZFS a usuários não-privilegiados das seguintes formas:

- As permissões individuais podem ser concedidas a um usuário, grupo ou todos.
- Os grupos de permissões individuais podem ser concedidos como um *conjunto de permissões* a um usuário, grupo ou todos.
- As permissões podem ser concedidas tanto localmente apenas ao conjunto de dados atual quanto a todos os descendentes do conjunto de dados atual.

A tabela abaixo descreve as operações que podem ser delegadas e as permissões dependentes requeridas para realizar as operações delegadas.

Permissão (Subcomando)	Descrição	Dependências
<code>allow</code>	A capacidade que você tem de conceder permissões a outro usuário.	Também deve ter a permissão que está sendo concedida.
<code>clone</code>	A capacidade de clonar quaisquer instantâneos do conjunto de dados.	Também deve possuir as capacidades <code>create</code> e <code>mount</code> no sistema de arquivos original.
<code>create</code>	A capacidade de criar conjuntos de dados descendentes.	Também deve possuir a capacidade <code>mount</code> .
<code>destroy</code>	A capacidade de destruir um conjunto de dados.	Também deve possuir a capacidade <code>mount</code> .
<code>mount</code>	A capacidade de montar e desmontar um conjunto de dados e de criar e destruir os links do dispositivo de volume.	
<code>promote</code>	A capacidade de promover um clone de um conjunto de dados.	Também deve possuir as capacidades <code>mount</code> e <code>promote</code> no sistema de arquivos original.
<code>receive</code>	A capacidade de criar sistemas de arquivos descendentes com o comando <code>zfs receive</code> .	Também deve possuir as capacidades <code>mount</code> e <code>create</code> .
<code>rename</code>	A capacidade de renomear um conjunto de dados.	Também deve possuir as capacidades <code>create</code> e <code>mount</code> no novo pai.
<code>reversão</code>	A capacidade de reverter um instantâneo.	
<code>send</code>	A capacidade de enviar um fluxo de instantâneo.	
<code>share</code>	A capacidade de compartilhar e cancelar o compartilhamento de um conjunto de dados.	
<code>instantâneo</code>	A capacidade de obter um instantâneo de um conjunto de dados.	

Você pode delegar os seguintes conjuntos de permissões, mas a permissão pode estar limitada para a permissão de acesso, leitura ou alteração:

- `groupquota`
- `groupused`
- `userprop`
- `userquota`
- `userused`

Além disso, você pode delegar as seguintes propriedades do ZFS a usuários não raiz:

- `aclinherit`
- `aclmode`
- `atime`
- `canmount`
- `casesensitivity`
- soma de verificação
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `nbmand`
- `normalization`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`
- `refreservation`
- `reservation`
- `secondarycache`
- `setuid`
- `shareiscsi`
- `sharenfs`
- `sharesmb`
- `snapdir`
- `utf8only`
- `version`
- `volblocksize`
- `volsize`
- `vscan`
- `xattr`
- `zoned`

Algumas destas propriedades podem ser definidas somente durante a criação do conjunto de dados. Para obter uma descrição destas propriedades, consulte [“Introduzindo as propriedades do ZFS” na página 187](#).

Delegação de permissões do ZFS (`zfs allow`)

A sintaxe de `zfs allow` é a seguinte:

```
zfs allow [-ldugecs] everyone|user|group[,...] perm[@setname,...] filesystem| volume
```

A sintaxe de `zfs allow` abaixo (em negrito) identifica a quem as permissões são delegadas:

```
zfs allow [-uge] | user | group | everyone [, ...] filesystem | volume
```

Podem ser especificadas várias entidades em uma lista separada por vírgulas. Se não for especificada nenhuma opção `-uge`, o argumento é interpretado preferencialmente como a palavra-chave `everyone`, a seguir, como um nome de usuário e, por último, como um nome de grupo. Para especificar um usuário ou grupo denominado “everyone”, use a opção `-u` ou `-g`. Para especificar um grupo com o mesmo nome de um usuário, use a opção `-g`. A opção `-c` concede as permissões no momento da criação.

A sintaxe de `zfs allow` abaixo (em negrito) identifica como as permissões e os conjuntos de permissões são especificados:

```
zfs allow [-s] ... perm | @setname [, ...] filesystem | volume
```

Podem ser especificadas várias permissões em uma lista separada por vírgulas. Os nomes das permissões são os mesmo dos subcomandos e propriedades do ZFS. Para obter mais informações, consulte a seção anterior.

As permissões podem ser reunidas em *conjuntos de permissões* e são definidas pela opção `-s`. Os conjuntos de permissões podem ser usados por outros comandos `zfs allow` no sistema de arquivos especificado e seus descendentes. Os conjuntos de permissões são avaliados dinamicamente, de modo que as alterações no conjunto são atualizadas imediatamente. Os conjuntos de permissões seguem as mesmas convenções de nomeação dos sistemas de arquivos ZFS, mas o nome deve começar com o sinal de arroba (`@`) e não pode ter mais de 64 caracteres.

A sintaxe de `zfs allow` abaixo (em negrito) identifica como as permissões são delegadas:

```
zfs allow [-ld] ... .. filesystem | volume
```

A opção `-l` indica que a permissão é concedida ao conjunto de dados especificado e não a seus descendentes, a menos que a opção `-d` também seja especificada. A opção `-d` indica que a permissão é concedida aos conjuntos de dados descendentes e não a este conjunto de dados, a menos que a opção `-l` também seja especificada. Se nenhuma das opções `-ld` forem especificadas, as permissões são concedidas ao sistema de arquivos ou volume e a todos os seus descendentes.

Removendo as permissões delegadas do ZFS (`zfs unallow`)

Você pode remover as permissões concedidas anteriormente com o comando `zfs unallow`.

Por exemplo, suponha que você delegou as permissões `create`, `destroy`, `mount` e `snapshot` como a seguir:

```
# zfs allow cindys create,destroy,mount,snapshot tank/cindys
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
      user cindys create,destroy,mount,snapshot
-----
```

Para remover estas permissões, pode ser necessário usar uma sintaxe semelhante à seguinte:

```
# zfs unallow cindys tank/cindys
# zfs allow tank/cindys
```

Usando a administração delegada de ZFS

Esta seção oferece exemplos de delegação e exibição de permissões delegadas do ZFS.

Delegando permissões do ZFS (Exemplos)

EXEMPLO 9-1 Delegando permissões a um usuário individual

Ao fornecer as permissões `create` e `mount` a um usuário individual, você precisa ter certeza de que o usuário tem as permissões no ponto de montagem subjacente.

Por exemplo, para dar ao usuário `marks` as permissões `create` e `mount` em `tank`, defina primeiro as permissões:

```
# chmod A+user:marks:add_subdirectory:fd:allow /tank
```

A seguir, use o comando `zfs allow` para conceder as permissões `create`, `destroy` e `mount`. Por exemplo:

```
# zfs allow marks create,destroy,mount tank
```

Agora, o usuário `marks` pode criar seus próprios sistemas de arquivos no sistema de arquivos `tank`. Por exemplo:

```
# su marks
marks$ zfs create tank/marks
marks$ ^D
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied
```

EXEMPLO 9-2 Delegando as permissões de criação e destruição a um grupo

O exemplo abaixo mostra como configurar um sistema de arquivos para que todos no grupo `staff` possam criar e montar sistemas de arquivos no sistema de arquivos `tank`, bem como destruir seus próprios sistemas de arquivos. No entanto, os membros do grupo `staff` não podem destruir os sistemas de arquivos de outros.

EXEMPLO 9-2 Delegando as permissões de criação e destruição a um grupo (Continuação)

```

# zfs allow staff create,mount tank
# zfs allow -c create,destroy tank
# zfs allow tank
-----
Create time permissions on (tank)
    create,destroy
Local+Descendent permissions on (tank)
    group staff create,mount
-----
# su cindys
cindys% zfs create tank/cindys
cindys% exit
# su marks
marks% zfs create tank/marks/data
marks% exit
cindys% zfs destroy tank/marks/data
cannot destroy 'tank/mark': permission denied

```

EXEMPLO 9-3 Delegando permissões no nível correto do sistema de arquivos

Certifique-se de ter concedido a permissão aos usuários no nível correto do sistema de arquivos. Por exemplo, ao usuário marks são concedidas as permissões create, destroy e mount para os sistemas de arquivos locais e descendentes. Ao usuário marks é concedida a permissão local para realizar instantâneos do sistema de arquivos tank, mas não tem permissão para realizar instantâneos do seu próprio sistema de arquivos. Assim, ele não tem a permissõesnapshot no nível correto do sistema de arquivos.

```

# zfs allow -l marks snapshot tank
# zfs allow tank
-----
Local permissions on (tank)
    user marks snapshot
Local+Descendent permissions on (tank)
    user marks create,destroy,mount
-----
# su marks
marks$ zfs snapshot tank/@snap1
marks$ zfs snapshot tank/marks@snap1
cannot create snapshot 'mark/marks@snap1': permission denied

```

Para conceder ao usuário marks a permissão no nível do descendente, use a opção zfs allow -d. Por exemplo:

```

# zfs unallow -l marks snapshot tank
# zfs allow -d marks snapshot tank
# zfs allow tank
-----
Descendent permissions on (tank)
    user marks snapshot
Local+Descendent permissions on (tank)
    user marks create,destroy,mount
-----

```

EXEMPLO 9-3 Delegando permissões no nível correto do sistema de arquivos *(Continuação)*

```
# su marks
$ zfs snapshot tank@snap2
cannot create snapshot 'tank@snap2': permission denied
$ zfs snapshot tank/marks@snappy
```

Agora, o usuário marks pode somente criar um instantâneo abaixo do nível tank.

EXEMPLO 9-4 Definindo e usando permissões delegadas complexas

Você pode conceder permissões específicas a usuários ou grupos. Por exemplo, o comando `zfs allow` seguinte concede permissões específicas ao grupo `staff`. Além disso, as permissões `destroy` e `snapshot` são concedidas após a criação dos sistemas de arquivos tank.

```
# zfs allow staff create,mount tank
# zfs allow -c destroy,snapshot tank
# zfs allow tank
```

```
-----
Create time permissions on (tank)
    destroy,snapshot
Local+Descendent permissions on (tank)
    group staff create,mount
-----
```

Por ser um membro do grupo `staff`, o usuário marks pode criar sistemas de arquivos tank. Além disso, o usuário marks pode criar um instantâneo de `tank/marks2` porque possui as permissões específicas para isso. Por exemplo:

```
# su marks
$ zfs create tank/marks2
$ zfs allow tank/marks2
-----
Local permissions on (tank/marks2)
    user marks destroy,snapshot
-----
Create time permissions on (tank)
    destroy,snapshot
Local+Descendent permissions on (tank)
    group staff create
    everyone mount
-----
```

Porém, ele não pode criar um instantâneo em `tank/marks` porque não possui as permissões específicas para isso. Por exemplo:

```
$ zfs snapshot tank/marks2@snap1
$ zfs snapshot tank/marks@snapp
cannot create snapshot 'tank/marks@snapp': permission denied
```

Se tiver a permissão `create` no diretório de início, você pode criar seus próprios diretórios de instantâneos. Esta situação é útil quando o sistema de arquivos está montado em NFS. Por exemplo:

EXEMPLO 9-4 Definindo e usando permissões delegadas complexas (Continuação)

```

$ cd /tank/marks2
$ ls
$ cd .zfs
$ ls
snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x  2 marks  staff          2 Dec 15 13:53 snap1
$ pwd
/tank/marks2/.zfs/snapshot
$ mkdir snap2
$ zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank                264K  33.2G  33.5K  /tank
tank/marks          24.5K  33.2G  24.5K  /tank/marks
tank/marks2         46K   33.2G  24.5K  /tank/marks2
tank/marks2@snap1  21.5K  -      24.5K  -
tank/marks2@snap2  0      -      24.5K  -
$ ls
snap1 snap2
$ rmdir snap2
$ ls
snap1

```

EXEMPLO 9-5 Definindo e usando um conjunto de permissões delegadas do ZFS

O exemplo seguinte mostra como criar um conjunto de permissões `@myset` e conceder e renomear permissões no grupo `staff` para o sistema de arquivos `tank`. O usuário `cindys`, membro do grupo `staff`, possui a permissão para criar um sistema de arquivos em `tank`. No entanto, o usuário `lp` não possui a permissão para criar um sistema de arquivos em `tank`.

```

# zfs allow -s @myset create,destroy,mount,snapshot,promote,clone,readonly tank
# zfs allow tank
-----
Permission sets on (tank)
    @myset clone,create,destroy,mount,promote,readonly,snapshot
-----
# zfs allow staff @myset,rename tank
# zfs allow tank
-----
Permission sets on (tank)
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions on (tank)
    group staff @myset,rename
# chmod A+group:staff:add_subdirectory:fd:allow tank
# su cindys
cindys% zfs create tank/data
Cindys% zfs allow tank
-----
Permission sets on (tank)
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions on (tank)
    group staff @myset,rename

```

EXEMPLO 9-5 Definindo e usando um conjunto de permissões delegadas do ZFS (Continuação)

```
-----
cindys% ls -l /tank
total 15
drwxr-xr-x  2 cindys  staff          2 Aug  8 14:10 data
cindys% exit
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied
```

Exibindo permissões delegadas do ZFS (Exemplos)

Você pode usar o comando seguinte para exibir permissões:

```
# zfs allow dataset
```

Este comando exibe permissões que são definidas ou concedidas a este conjunto de dados. A saída contém os seguintes componentes:

- Conjuntos de permissões
- Permissões específicas ou permissões no momento da criação
- Conjunto de dados local
- Conjuntos de dados locais e descendentes
- Conjuntos de dados descendentes somente

EXEMPLO 9-6 Exibindo permissões básicas de administração delegada

A saída do exemplo seguinte indica que o usuário `cindys` tem permissões para criar, destruir, montar e realizar um instantâneo no sistema de arquivos `tank/cindys`.

```
# zfs allow tank/cindys
```

```
-----
Local+Descendent permissions on (tank/cindys)
user cindys create,destroy,mount,snapshot
```

EXEMPLO 9-7 Exibindo permissões complexas de administração delegada

A saída deste exemplo indica as seguintes permissões nos sistemas de arquivos `pool/fred` e `pool`.

Para o sistema de arquivos `pool/fred`:

- São definidas dois conjuntos de permissões:
 - `@eng` (create, destroy, snapshot, mount, clone, promote, rename)
 - `@simple` (create, mount)
- As permissões do tempo da criação são definidas para o conjunto de permissões `@eng` e a propriedade `mountpoint`. Tempo de criação significa que o conjunto de permissões `@eng` e a propriedade `mountpoint` são concedidas depois da criação do conjunto de dados.

EXEMPLO 9-7 Exibindo permissões complexas de administração delegada (Continuação)

- Ao usuário `tom` é concedido o conjunto de permissões `@eng` e ao usuário `joe` são concedidas as permissões `create`, `destroy` e `mount` para sistemas de arquivos locais.
- Ao usuário `fred` é concedido o conjunto de permissões `@basic` e as permissões `share` e `rename` para os sistemas de arquivos locais e descendentes.
- Ao usuário `barney` e ao grupo `staff` é concedido o conjunto de permissões `@basic` somente para os sistemas de arquivos descendentes.

Para o sistema de arquivos `pool`:

- Está definido o conjunto de permissões `@simple` (`create`, `destroy`, `mount`).
- Ao grupo `staff` é concedido o conjunto de permissões `@simple` no sistema de arquivos local.

Eis a saída deste exemplo:

```
$ zfs allow pool/fred
-----
Permission sets on (pool/fred)
    @eng create,destroy,snapshot,mount,clone,promote,rename
    @simple create,mount
Create time permissions on (pool/fred)
    @eng,mountpoint
Local permissions on (pool/fred)
    user tom @eng
    user joe create,destroy,mount
Local+Descendent permissions on (pool/fred)
    user fred @basic,share,rename
Descendent permissions on (pool/fred)
    user barney @basic
    group staff @basic
-----
Permission sets on (pool)
    @simple create,destroy,mount
Local permissions on (pool)
    group staff @simple
-----
```

Removendo permissões do ZFS (Exemplos)

Você pode usar o comando `zfs unallow` para remover as permissões concedidas. Por exemplo, o usuário `cindys` tem permissões para criar, destruir, montar e realizar um instantâneo no sistema de arquivos `tank/cindys`.

```
# zfs allow cindys create,destroy,mount,snapshot tank/cindys
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
```

```
user cindys create,destroy,mount,snapshot
```

A sintaxe seguinte de `zfs unallow` remove a permissão de criação de instantâneo do usuário `cindys` do sistema de arquivos `tank/cindys`:

```
# zfs unallow cindys snapshot tank/cindys
# zfs allow tank/cindys
-----
Local+Descendent permissions on (tank/cindys)
    user cindys create,destroy,mount
-----
cindys% zfs create tank/cindys/data
cindys% zfs snapshot tank/cindys@today
cannot create snapshot 'tank/cindys@today': permission denied
```

Como no outro o exemplo, o usuário `marks` tem as seguintes permissões em `tank/marks`:

```
# zfs allow tank/marks
-----
Local+Descendent permissions on (tank/marks)
    user marks create,destroy,mount
-----
```

Neste exemplo, a sintaxe seguinte de `zfs unallow` remove todas as permissões do usuário `marks` de `tank/marks`:

```
# zfs unallow marks tank/marks
```

A sintaxe seguinte de `zfs unallow` remove um conjunto de permissões do sistema de arquivos `tank`.

```
# zfs allow tank
-----
Permission sets on (tank)
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Create time permissions on (tank)
    create,destroy,mount
Local+Descendent permissions on (tank)
    group staff create,mount
-----
# zfs unallow -s @myset tank
$ zfs allow tank
-----
Create time permissions on (tank)
    create,destroy,mount
Local+Descendent permissions on (tank)
    group staff create,mount
-----
```

Tópicos avançados do ZFS do Oracle Solaris

Este capítulo descreve os volumes ZFS, o uso do ZFS em um sistema Solaris com regiões instaladas, pools de raiz alternativa e perfis de direitos do ZFS.

Este capítulo traz as seguintes seções:

- “Volumes ZFS” na página 277
- “Usando ZFS em um sistema Solaris com regiões instaladas” na página 280
- “Usando pools de raiz alternativa do ZFS” na página 286
- “Perfis de direitos do ZFS” na página 287

Volumes ZFS

Um volume ZFS consiste em um conjunto de dados que representa um dispositivo de bloco. Os volumes do ZFS são identificados como dispositivos no diretório `/dev/zvol/{dsk, rdsk}/pool`.

No exemplo a seguir, um volume do ZFS de 5 GB, `tank/vol` é criado:

```
# zfs create -V 5gb tank/vol
```

Ao criar um volume, uma reserva é automaticamente definida ao tamanho inicial do volume para evitar a ocorrência de comportamento inesperado. Por exemplo, se o tamanho do volume diminuir, os dados podem se corromper. Você deve ter cuidado ao alterar o tamanho do volume.

Além disso, se criar um instantâneo de um volume que altera em tamanho, você pode introduzir inconsistências se tentar reverter o instantâneo ou criar um clone do instantâneo.

Para obter informações sobre as propriedades do sistema de arquivos que podem ser aplicadas a volumes, consulte a [Tabela 6-1](#).

Se estiver usando um sistema Solaris com regiões instaladas, você não pode criar ou clonar um volume ZFS em uma região não global. Qualquer tentativa para tal irá falhar. Para obter informações sobre o uso de volumes ZFS em uma região global, consulte [“Adicionando volumes ZFS a uma região não global” na página 282.](#)

Usando um volume ZFS como um dispositivo de intercâmbio ou de dump

Durante a instalação de um sistema de arquivos raiz do ZFS ou a migração de um sistema de arquivos raiz do UFS, é criado um dispositivo de permuta em um volume do ZFS no conjunto raiz do ZFS. Por exemplo:

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

Durante a instalação de um sistema de arquivos raiz do ZFS ou a migração de um sistema de arquivos raiz do UFS, é criado um dispositivo de despejo em um volume do ZFS no conjunto raiz do ZFS. O dispositivo de despejo não requer administração depois de configurado. Por exemplo:

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

Se precisar alterar a área de intercâmbio ou o dispositivo de dump após a instalação ou atualização do sistema, use os comandos `swap` e `dumpadm` como nas versões anteriores do Solaris. Se precisar criar um volume de permuta adicional, crie um volume do ZFS de um tamanho específico e ative a permuta nesse dispositivo. Por exemplo:

```
# 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
```

Não permuta para um arquivo em um sistema de arquivos ZFS. A configuração de arquivos de intercâmbio ZFS não é suportada.

Para obter informações sobre o ajuste do tamanho dos volumes de permuta e despejo, consulte [“Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS” na página 166.](#)

Usando um volume ZFS como um destino iSCSI Solaris

É possível criar facilmente um volume do ZFS como um destino iSCSI, definindo a propriedade `shareiscsi` no volume. Por exemplo:

```
# zfs create -V 2g tank/volumes/v2
# zfs set shareiscsi=on tank/volumes/v2
# iscsitadm list target
Target: tank/volumes/v2
  iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
  Connections: 0
```

Depois da criação do destino iSCSI, configure o iniciador iSCSI. Para obter mais informações sobre iniciadores e destinos iSCSI Solaris, consulte o [Capítulo 14, “Configuring Oracle Solaris iSCSI Targets and Initiators \(Tasks\),”](#) no *System Administration Guide: Devices and File Systems*.

Observação – Os destinos iSCSI Solaris também podem ser criados e gerenciados com o comando `iscsitadm`. Se definir a propriedade `shareiscsi` em um volume ZFS, não use o comando `iscsitadm` para criar o mesmo dispositivo de destino. Caso contrário, você cria informações de destino duplicadas para o mesmo dispositivo.

Um volume do ZFS, como um destino iSCSI, é gerenciado da mesma forma que qualquer outro conjunto de dados do ZFS. No entanto, as operações `rename`, `export` e `import` funcionam um pouco diferente nos destinos iSCSI.

- Quando um volume ZFS é renomeado, o nome do destino iSCSI permanece o mesmo. Por exemplo:

```
# zfs rename tank/volumes/v2 tank/volumes/v1
# iscsitadm list target
Target: tank/volumes/v1
  iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
  Connections: 0
```

- A exportação de um pool que contém um volume ZFS compartilhado faz com que o destino seja removido. A importação de um pool que contém um volume ZFS compartilhado faz com que o destino seja compartilhado. Por exemplo:

```
# zpool export tank
# iscsitadm list target
# zpool import tank
# iscsitadm list target
Target: tank/volumes/v1
  iSCSI Name: iqn.1986-03.com.sun:02:984fe301-c412-ccc1-cc80-cf9a72aa062a
  Connections: 0
```

Todas as informações de configuração do destino iSCSI são armazenadas com o conjunto de dados. Assim como um sistema de arquivos compartilhado NFS, um destino iSCSI que é importado em um sistema diferente é compartilhado apropriadamente.

Usando ZFS em um sistema Solaris com regiões instaladas

As seções a seguir descrevem como utilizar o ZFS em um sistema com regiões Oracle Solaris:

- “Adicionando sistemas de arquivos ZFS a uma região não global” na página 281
- “Delegando conjuntos de dados a uma região não global” na página 282
- “Adicionando volumes ZFS a uma região não global” na página 282
- “Usando pools de armazenamento do ZFS dentro de uma região” na página 283
- “Gerenciando propriedades do ZFS dentro de uma região” na página 283
- “Entendendo a propriedade zoned” na página 284

Para obter informações sobre a configuração de regiões em um sistema com um sistema de arquivos raiz do ZFS que será migrado ou terá patches instalados com o Oracle Solaris Live Upgrade, consulte “Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (Solaris 10 10/08)” na página 149 ou “Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com regiões (no mínimo Solaris 10 5/09)” na página 155.

Leve em consideração os seguintes pontos ao associar os conjuntos de dados ZFS a regiões:

- Você pode adicionar um sistema de arquivos ou clone do ZFS a uma região não global com ou sem delegar o controle administrativo.
- É possível adicionar um volume do ZFS como um dispositivo em regiões não globais.
- No momento, não é possível associar instantâneos do ZFS a regiões.

Nas seções a seguir, um conjunto de dados do ZFS refere-se a um sistema de arquivos ou um clone.

A adição de um conjunto de dados permite que a região não global compartilhe espaço em disco com a região global, embora o administrador de região não possa controlar as propriedades ou criar novos sistemas de arquivos na hierarquia do sistema de arquivos subjacente. Esta operação é idêntica à adicionar outro tipo de sistema de arquivos a uma região, e deve ser utilizada quando o objetivo principal for unicamente compartilhar espaço em disco comum.

O ZFS também permite que os conjuntos de dados sejam delegados a uma região não global, proporcionando controle total do administrador de regiões sobre o conjunto de dados e seus filhos. O administrador de região pode criar e destruir os sistemas de arquivos ou clones dentro deste conjunto de dados, assim como modificar propriedades dos conjuntos de dados. O administrador de regiões não pode afetar os conjuntos de dados que não foram adicionados à região, o que inclui exceder as cotas máximas definidas no conjunto de dados delegado.

Considere as seguintes interações ao trabalhar com o ZFS em um sistema Oracle Solaris com regiões instaladas:

- Um sistema de arquivos ZFS adicionado a uma região não global deve ter a propriedade mountpoint definida como legado.

- Devido ao problema 6449301, não adicione um conjunto de dados ZFS em uma região não global quando a região não global estiver configurada. Em vez disso, adicione um conjunto de dados ZFS depois que região for instalada.
- Quando o `zonepath` de fonte e o `zonepath` de destino residirem em um sistema de arquivos do ZFS e estiverem no mesmo conjunto, o comando `zoneadm clone` utilizará automaticamente o ZFS para clonar uma região. O comando `zoneadm clone` criará um instantâneo do ZFS do `zonepath` de fonte e configurará o `zonepath` de destino. Não é possível usar o comando `zfs clone` para clonar uma região. Para obter mais informações, consulte [Parte II, “Regiões,” no Guia de administração do sistema: gerenciamento de recursos Oracle Solaris Containers e Oracle Solaris Zones.](#)
- Caso você delegue um sistema de arquivos do ZFS para uma região não global, é preciso remover aquele sistema de arquivos da região não global antes de utilizar o Oracle Solaris Live Upgrade. Caso contrário, o Oracle Live Upgrade irá falhar devido a um erro de sistema de arquivos de somente leitura.

Adicionando sistemas de arquivos ZFS a uma região não global

Você pode adicionar um sistema de arquivos ZFS como um sistema de arquivos genérico quando o objetivo for unicamente compartilhar espaço com a região global. Um sistema de arquivos ZFS adicionado a uma região não global deve ter a propriedade `mountpoint` definida como `legacy`.

O sistema de arquivos ZFS pode ser adicionado a uma região não global com o subcomando `add fs` do comando `zonecfg`.

No exemplo seguinte, um sistema de arquivos do ZFS é adicionado a uma região não global por um administrador global na região global:

```
# zonecfg -z zion
zonecfg:zion> add fs
zonecfg:zion:fs> set type=zfs
zonecfg:zion:fs> set special=tank/zone/zion
zonecfg:zion:fs> set dir=/export/shared
zonecfg:zion:fs> end
```

Esta sintaxe adiciona o sistema de arquivos do ZFS, `tank/zone/zion`, à região `zion` já configurada, montada em `/export/shared`. A propriedade `mountpoint` do sistema de arquivos deve ser definida como `legacy` (legado), e o sistema de arquivos já não poderá ser montado em outro local. O administrador de regiões pode criar e destruir arquivos dentro do sistema de arquivos. O sistema de arquivos não pode ser remontado em uma localização diferente e o administrador de regiões não pode alterar propriedades no sistema de arquivos, como `atime`, `readonly`, `compression`, e etc. O administrador de regiões globais é responsável por definir e controlar as propriedades do sistema de arquivos.

Para obter mais informações sobre o comando `zonecfg` e sobre a configuração dos tipos de recursos com `zonecfg`, consulte [Parte II, “Regiões,” no Guia de administração do sistema: gerenciamento de recursos Oracle Solaris Containers e Oracle Solaris Zones](#).

Delegando conjuntos de dados a uma região não global

Se a prioridade for delegar a administração do armazenamento a uma região, o ZFS oferece suporte à adição de conjuntos de dados a uma região não global através do subcomando `add dataset` do comando `zonecfg`.

No exemplo a seguir, um sistema de arquivos do ZFS é delegado a uma região não global por um administrador global na região global.

```
# zonecfg -z zion
zonecfg:zion> add dataset
zonecfg:zion:dataset> set name=tank/zone/zion
zonecfg:zion:dataset> end
```

Ao contrário da adição de um sistema de arquivos, esta sintaxe faz com que o sistema de arquivos ZFS `tank/zone/zion` esteja visível dentro da região `zion` já configurada. O administrador de regiões pode definir as propriedades do sistema de arquivos e criar sistemas de arquivos descendentes. Além disso, o administrador de região pode criar instantâneos, clones e, por outro lado, controlar toda a hierarquia do sistema de arquivos.

Se estiver utilizando o Oracle Solaris Live Upgrade para atualizar o BE do ZFS com regiões não globais, remova primeiro todos os conjuntos de dados delegados. Caso contrário, Oracle Solaris Live Upgrade falhará e apresentará o erro de sistema de arquivos de somente leitura. Por exemplo:

```
zonecfg:zion>
zonecfg:zion> remove dataset name=tank/zone/zion
zonecfg:zion1> exit
```

Para obter mais informações sobre que ações são permitidas dentro das regiões, consulte [“Gerenciando propriedades do ZFS dentro de uma região” na página 283](#).

Adicionando volumes ZFS a uma região não global

Os volumes ZFS não podem ser adicionados a uma região não global com o subcomando `add dataset` do comando `zonecfg`. No entanto, os volumes podem ser adicionados a uma região com o subcomando `add device` do comando `zonecfg`.

No exemplo a seguir, um volume do ZFS é adicionado a uma região não global por um administrador de região global na região global:

```
# zonecfg -z zion
zion: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zion> create
zonecfg:zion> add device
zonecfg:zion:device> set match=/dev/zvol/dsk/tank/vol
zonecfg:zion:device> end
```

Esta sintaxe adiciona o volume tank/vol para a região zion . Observe que a adição de um volume bruto a uma região apresenta riscos de segurança implícitos, mesmo que o volume não corresponda a um dispositivo físico. Em particular, o administrador de regiões poderia criar sistemas de arquivos inválidos que causariam graves danos no sistema quando houvesse uma tentativa de montagem. Para obter mais informações sobre a adição de dispositivos a regiões e os riscos de segurança relacionados, consulte [“Entendendo a propriedade zoned” na página 284.](#)

Para obter mais informações sobre a adição de dispositivos a regiões, consulte [Parte II, “Regiões,” no Guia de administração do sistema: gerenciamento de recursos Oracle Solaris Containers e Oracle Solaris Zones.](#)

Usando pools de armazenamento do ZFS dentro de uma região

Os pools de armazenamento do ZFS não podem ser criados ou modificados dentro de uma região. O modelo de administração delegado centraliza o controle dos dispositivos de armazenamento físico dentro da região global e o controle do armazenamento virtual em regiões não globais. Enquanto um conjunto de dados no nível do conjunto puder ser adicionado a uma região, nenhum comando que modifique as características físicas do conjunto, como criar, adicionar ou remover dispositivos, poderá ser realizado de dentro de uma região. Mesmo se os dispositivos físicos forem adicionados a uma região com o subcomando add device do comando zonecfg, ou se os arquivos forem usados, o comando zpool não permite a criação de novos pools dentro da região.

Gerenciando propriedades do ZFS dentro de uma região

Depois que o conjunto de dados for delegado a uma região, o administrador de região poderá controlar propriedades específicas do conjunto de dados. Depois que um conjunto de dados é delegado a uma região, todos os seus antepassados estão visíveis como conjuntos de dados de somente leitura, enquanto que o próprio conjunto de dados é gravável, assim como seus descendentes. Considere, por exemplo, a seguinte configuração:

```
global# zfs list -Ho name
tank
tank/home
```

```
tank/data
tank/data/matrix
tank/data/zion
tank/data/zion/home
```

Se `tank/data/zion` fosse adicionado a uma região, cada conjunto de dados teria as seguintes propriedades.

Conjunto de dados	Visível	Gravável	Propriedades imutáveis
<code>tank</code>	Sim	Não	-
<code>tank/home</code>	Não	-	-
<code>tank/data</code>	Sim	Não	-
<code>tank/data/matrix</code>	Não	-	-
<code>tank/data/zion</code>	Sim	Sim	<code>sharenfs, zoned, quota, reservation</code>
<code>tank/data/zion/home</code>	Sim	Sim	<code>sharenfs, zoned</code>

Observe que todos os pais de `tank/zone/zion` estão visíveis como somente leitura, todos os descendentes são graváveis e os conjuntos de dados que não fazem parte da hierarquia pai não estão visíveis de nenhuma forma. O administrador de região não pode alterar a propriedade `sharenfs` porque as regiões não globais não podem agir como servidores NFS. O administrador de região não pode alterar a propriedade `zoned` porque isso apresentaria um risco de segurança, conforme descrito na próxima seção.

Os usuários privilegiados na região podem alterar qualquer outra propriedade que possa ser definida, exceto as propriedades `quota` e `reservation`. Este comportamento permite que o administrador de região global controle o consumo do espaço em disco de todos os conjuntos de dados utilizados pela região não global.

Além disso, as propriedades `sharenfs` e `mountpoint` não podem ser alteradas pelo administrador de região global depois que um conjunto de dados tiver sido delegado a uma região não global.

Entendendo a propriedade `zoned`

Quando um conjunto de dados é delegado a uma região não global, ele deve ser identificado com uma marca especial para que determinadas propriedades não sejam interpretadas dentro do contexto da região global. Depois que o conjunto de dados tiver sido delegado a uma região não global e estiver sob o controle de um administrador de região, já não será possível confiar em seu conteúdo. Como em qualquer sistema de arquivos, podem haver binários `setuid`, links simbólicos ou, senão, conteúdos duvidosos que poderiam afetar a segurança da região global.

Além disso, a propriedade `mountpoint` não pode ser interpretada no contexto da região global. Caso contrário, o administrador de regiões poderia afetar o espaço de nome da região global. Para abordar este último, o ZFS usa a propriedade `zoned` para indicar que um conjunto de dados foi delegado a uma região não global em um dado ponto no tempo.

A propriedade `zoned` é um valor booleano ativado automaticamente quando uma região, contendo um conjunto de dados ZFS, é inicializada pela primeira vez. Um administrador de regiões não precisa ativar esta propriedade manualmente. Se a propriedade `zoned` estiver definida, o conjunto de dados não pode ser montado ou compartilhado na região global. No exemplo a seguir, o `tank/zone/zion` foi delegado a uma região, enquanto o `tank/zone/global`, não:

```
# 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
```

Observe a diferença entre a propriedade `mountpoint` e o diretório no qual o conjunto de dados `tank/zone/zion` está montado atualmente. A propriedade `mountpoint` reflete a propriedade como estando armazenada em disco, não onde o conjunto de dados está montado atualmente no sistema.

Quando um conjunto de dados é removido de uma região ou quando uma região é destruída, a propriedade `zoned` não é apagada automaticamente. Este comportamento se deve aos riscos de segurança inerentes relacionados a essas tarefas. Devido ao fato de um usuário não confiável ter tido total acesso ao conjunto de dados e a seus descendentes, a propriedade `mountpoint` pode ser definida com valores incorretos ou pode haver binários `setuid` nos sistemas de arquivos.

Para evitar eventuais riscos de segurança, a propriedade `zoned` deve ser apagada manualmente pelo administrador global se você desejar reutilizar o conjunto de dados. Antes de definir a propriedade `zoned` como `off`, certifique-se de que a propriedade `mountpoint` do conjunto de dados e de todos os seus descendentes esteja definida com valores aceitáveis e que não existam binários `setuid`, ou que a propriedade `setuid` esteja desativada.

Depois de ter verificado se não existem vulnerabilidades na segurança, a propriedade `zoned` pode ser desativada através do comando `zfs set` ou `zfs inheri`. Se a propriedade `zoned` for desativada enquanto um conjunto de dados estiver sendo usado dentro de uma região, o sistema pode se comportar de forma imprevisível. Altere a propriedade somente se tiver certeza de que o conjunto de dados já não esteja sendo usado por uma região não global.

Usando pools de raiz alternativa do ZFS

Quando um conjunto é criado, ele é vinculado ao sistema do host. O sistema do host mantém informações sobre o conjunto para poder detectar quando o conjunto não está disponível. Embora úteis em operações normais, estas informações podem se tornar obstáculos quando se está inicializando a partir de mídias alternativas ou criando um conjunto em uma mídia removível. Para resolver este problema, o ZFS oferece um recurso de pools de *raiz alternativa*. O pool de raiz alternativa não é preservado em reinicializações do sistema e todos os pontos de montagem são modificados para terem relação com a raiz do pool.

Criando pools de raiz alternativa do ZFS

O motivo mais comum para criar um conjunto raiz alternativo é para utilização com mídias removíveis. Nesse caso, os usuários querem normalmente um único sistema de arquivos e querem que tal sistema possa ser montado em qualquer parte do sistema de destino. Quando um conjunto raiz alternativo é criado com a opção `zpool create -R`, o ponto de montagem do sistema de arquivos raiz é automaticamente definido como `/`, que equivale à própria raiz alternativa.

No exemplo abaixo, um pool denominado `morpheus` é criado com `/mnt` como caminho de raiz alternativa:

```
# zpool create -R /mnt morpheus c0t0d0
# zfs list morpheus
NAME                USED  AVAIL  REFER  MOUNTPOINT
morpheus            32.5K 33.5G   8K    /mnt
```

Observe o sistema de arquivos único, `morpheus`, cujo ponto de montagem é a raiz alternativa do pool, `/mnt`. O ponto de montagem armazenado em disco é `/` e o caminho completo para `/mnt` é interpretado somente no contexto inicial da criação do conjunto. Este sistema de arquivos poderá, então, ser importado e exportado através de um conjunto raiz alternativo arbitrário em um sistema diferente utilizando a sintaxe do *valor de raiz alternativa* `-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
```

Importando pools de raiz alternativa

Os pools também podem ser importados usando uma raiz alternativa. Este recurso leva em consideração as situações de recuperação, nas quais os pontos de montagem não devem ser

interpretados no contexto da raiz atual, mas sim em algum diretório temporário no qual seja possível realizar correções. Este recurso também pode ser utilizado ao montar mídias removíveis conforme descrito anteriormente.

No exemplo abaixo, um pool denominado morpheus é importado com /mnt como o caminho de raiz alternativa. Neste exemplo pressupõe-se que morpheus foi previamente exportado.

```
# zpool import -R /a pool
# zpool list morpheus
NAME  SIZE  ALLOC  FREE    CAP  HEALTH  ALROOT
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
```

Perfis de direitos do ZFS

Se desejar efetuar as tarefas de gerenciamento do ZFS sem utilizar a conta de superusuário (raiz), é possível adotar uma função com um dos perfis seguintes para realizar as tarefas de administração do ZFS:

- Gerenciamento de armazenamento do ZFS: oferece o privilégio de criar, destruir e manipular dispositivos dentro de um conjunto de armazenamento do ZFS
- Gerenciamentos de sistema de arquivos do ZFS: oferece o privilégio de criar, destruir e modificar os sistemas de arquivos do ZFS

Para obter mais informações sobre a criação ou atribuição de funções, consulte o [System Administration Guide: Security Services](#).

Além de usar as funções RBAC para administrar sistemas de arquivos ZFS, também se deve levar em consideração o uso da administração delegada do ZFS nas tarefas de administração distribuídas do ZFS. Para obter mais informações, consulte o [Capítulo 9, “Administração delegada de ZFS”](#).

Solução de problemas e conjunto de recuperação do Oracle Solaris ZFS

Este capítulo descreve como identificar e recuperar a partir de falhas do ZFS. Também são fornecidas informações sobre como evitar tais falhas.

Este capítulo traz as seguintes seções:

- “Identificando falhas do ZFS” na página 289
- “Verificando a integridade do sistema de arquivos ZFS” na página 291
- “Resolvendo problemas com o ZFS” na página 293
- “Reparando uma configuração do ZFS danificada” na página 298
- “Reparando um dispositivo faltando” na página 299
- “Substituindo ou reparando um dispositivo modificado” na página 301
- “Reparando dados danificados” na página 310
- “Reparando um sistema não inicializável” na página 314

Identificando falhas do ZFS

Por ser uma combinação de um sistema de arquivos e um gerenciador de volumes, o ZFS pode exibir diferentes falhas. Este capítulo começa delineando as várias falhas e, em seguida, discute como identificá-las em um sistema em execução. E finalmente termina tratando o tema de como reparar os problemas. O ZFS pode encontrar três tipos básicos de erros:

- “Ausência de dispositivos em um pool de armazenamento do ZFS” na página 290
- “Dispositivos danificados em um pool de armazenamento do ZFS” na página 290
- “Dados corrompidos do ZFS” na página 290

Observe que um único pool pode sofrer os três tipos de erros, de modo que o procedimento completo de reparação implica em encontrar e corrigir o erro, passar para o próximo erro, e assim por diante.

Ausência de dispositivos em um pool de armazenamento do ZFS

Se um dispositivo for completamente removido do sistema, o ZFS detecta que o dispositivo não pode ser aberto e o coloca no estado REMOVIDO. Dependendo do nível de replicação dos dados do conjunto, essa remoção pode ou não fazer com que todo o conjunto se torne indisponível. Se, em um dispositivo RAID-Z ou espelhado, um disco for removido, o pool continua acessível. Um conjunto pode se tornar FAULTED, o que significa que nenhum dado é acessível até que o dispositivo seja desanexado, sob as condições a seguir:

- Se todos os componentes de um espelho são removidos
- Se mais de um dispositivo em um dispositivo (raidz1) RAID-Z é removido
- Se o dispositivo de nível superior é removido em uma configuração de disco único

Dispositivos danificados em um pool de armazenamento do ZFS

O termo “danificado” abrange uma ampla variedade de possíveis erros. Os exemplos incluem o seguinte:

- Erros transitórios de E/S devido a disco ou controlador defeituosos
- Corrupção de dados em disco devido a raios cósmicos
- Erros de driver resultando em transferência de dados de ou para locais incorretos
- Um usuário substitui porções do dispositivo físico por acidente

Em alguns casos, estes erros são transitórios, como um erro de E/S aleatório durante problemas com o controlador. Em outros casos, o problema pode ser permanente, como a corrupção em disco. Ainda assim, se o problema for permanente, isso não significa necessariamente que o erro ocorrerá novamente. Por exemplo, se um administrador substitui acidentalmente parte de um disco, e nenhum tipo de falha de hardware ocorre, o dispositivo não precisa ser trocado. Identificar exatamente o problema com o dispositivo não é uma tarefa fácil, por isso esse tema é abordado mais detalhadamente em uma seção posterior.

Dados corrompidos do ZFS

A corrupção de dados ocorre quando um ou mais erros no dispositivo (indicando um ou mais dispositivos ausentes ou danificados) afetam o dispositivo virtual de nível superior. Por exemplo, a metade de um espelho pode sofrer milhares de erros de dispositivo sem jamais causar corrupção de dados. Haverá corrupção de dados se for encontrado um erro no outro lado do espelho no mesmo exato local.

A corrupção de dados é sempre permanente e requer cuidados especiais durante a reparação. Mesmo que os dispositivos subjacentes forem reparados ou substituídos, os dados originais não poderão ser recuperados. Frequentemente, esse tipo de situação requer a recuperação dos dados a partir de backups. Os erros dos dados são registrados à medida que vão sendo encontrados e podem ser controlados através de scrubbing rotineira do conjunto, como explicado na seção seguinte. Quando um bloco corrompido é removido, o próximo ciclo de limpeza reconhece que a corrupção já não existe e remove qualquer vestígio de erro do sistema.

Verificando a integridade do sistema de arquivos ZFS

Não existe o utilitário `fsck` equivalente para o ZFS. Esse utilitário tem tradicionalmente servido a dois propósitos, ao reparo e à validação do sistema de arquivos.

Reparo de sistema de arquivos

Com os sistemas de arquivos tradicionais, a forma como os dados são gravados está inerentemente vulnerável a falhas inesperadas, provocando inconsistências de dados. Como um sistema de arquivos tradicional não é transacional, blocos não referenciados, contagens ruins de link ou outras estruturas de sistema de arquivos inconsistentes são possíveis. A adição de registros de ações resolve alguns destes problemas, porém pode trazer outros problemas quando o registro não puder ser revertido. A única forma para dados inconsistentes existirem no disco em uma configuração ZFS é através de falha do hardware (nesse caso o conjunto deve possuir redundância) ou quando um erro existir no software ZFS.

O utilitário `fsck` repara problemas conhecidos específicos para sistemas de arquivos UFS. A maioria dos problemas do conjunto de armazenamento do ZFS são relacionados com falhas de hardware ou falhas de energia. Muitos problemas podem ser evitados através da utilização de conjuntos redundantes. Se o conjunto for danificado por falhas de hardware ou queda de energia, consulte [“Reparando o dano de todo o pool de armazenamento do ZFS”](#) na página 313.

Se o conjunto não for redundante, haverá sempre a possibilidade de que a corrupção do sistema de arquivos torne alguns ou todos os dados inacessíveis.

Validação do sistema de arquivos

Além de efetuar a reparação do sistemas de arquivos, o utilitário `fsck` valida que os dados em disco não apresentam problemas. Tradicionalmente, essa tarefa requer a desmontagem do sistema de arquivos e a execução do utilitário `fsck`, levando o sistema possivelmente para o modo de usuário único no processo. Este quadro tem como resultado um tempo de inatividade proporcional ao tamanho do sistema de arquivos que está sendo verificado. Em vez de exigir um utilitário explícito para efetuar a verificação necessária, o ZFS fornece um mecanismo para

efetuar verificações rotineiras de todos as inconsistências. Esse recurso, conhecido como *scrubbing*, é utilizado, geralmente, na memória e em outros sistemas como um método de detecção e prevenção de erros, antes que estes provoquem falha de hardware ou de software.

Controlando o scrubbing de dados do ZFS

Sempre que o ZFS encontrar um erro através de scrubbing ou ao acessar um arquivo por demanda, o erro é registrado internamente para que você possa ter uma visão geral rápida de todos os erros conhecidos no conjunto.

Scrubbing explícito de dados do ZFS

A forma mais simples de verificar a integridade dos dados é iniciar um scrubbing explícito de todos os dados do conjunto. Esta operação percorre todos os dados do pool uma vez e comprova que todos os blocos possam ser lidos. O scrubbing se desenvolve tão rápido quanto os dispositivos permitam, sem bem que a prioridade de qualquer E/S é menor que a prioridade dada às operações normais. Essa operação pode impactar negativamente o desempenho, embora os dados do conjunto devam permanecer utilizáveis e um pouco receptivos enquanto o scrubbing ocorrer. Para iniciar um scrubbing explícito, use o comando `zpool scrub`. Por exemplo:

```
# zpool scrub tank
```

O status da operação de scrubbing atual pode ser exibido utilizando o comando `zpool status`. Por exemplo:

```
# zpool status -v tank
pool: tank
state: ONLINE
scrub: scrub completed after 0h7m with 0 errors on Tue Tue Feb  2 12:54:00 2010
config:
   NAME        STATE      READ WRITE CKSUM
   tank        ONLINE    0     0     0
     mirror-0  ONLINE    0     0     0
       c1t0d0  ONLINE    0     0     0
       c1t1d0  ONLINE    0     0     0

errors: No known data errors
```

Somente uma operação de scrubbing ativa por conjunto pode ocorrer por vez.

É possível parar uma operação de scrubbing em andamento com a opção `-s`. Por exemplo:

```
# zpool scrub -s tank
```

Na maioria dos casos, as operações de scrubbing devem chegar até o final para garantir a integridade dos dados. Interrompa uma operação de scrubbing de acordo com os seus próprios critérios se a performance do sistema é impactada pela operação.

Efetuar scrubbing de rotina garante E/S contínua para todos os discos do sistema. Scrubbing rotineiros apresentam a desvantagem de não permitir que o gerenciamento de energia coloque os discos inativos no modo de energia baixa. Se o sistema estiver geralmente realizando E/S sem parar ou se o consumo de energia não for uma preocupação, então esta questão pode ser ignorada sem perigo.

Para obter mais informações sobre a interpretação da saída `zpool status`, consulte [“Consultando status de pool de armazenamento do ZFS” na página 106](#).

Scrubbing e resilvering de dados do ZFS

Quando um dispositivo é substituído, inicia-se uma operação de resilvering para mover os dados provenientes de cópias boas para o novo dispositivo. Esta ação é uma forma de scrubbing de disco. Portanto, somente uma dessas ações pode ocorrer em um dado momento no conjunto. Se uma operação de scrubbing estiver em andamento, a operação de resilvering suspende o scrubbing atual e o reinicia depois que o resilvering for concluído.

Para obter mais informações sobre o resilvering, consulte [“Exibindo o status do resilvering” na página 309](#).

Resolvendo problemas com o ZFS

As seções a seguir descrevem como identificar e resolver problemas nos sistemas de arquivos ou nos conjuntos de armazenamento ZFS:

- [“Determinando se há problemas em um conjunto de armazenamento do ZFS” na página 294](#)
- [“Revisando a saída de `zpool status`” na página 295](#)
- [“Relatório de mensagens de erros do ZFS do sistema” na página 298](#)

Você pode usar os seguintes recursos para identificar problemas na configuração do seu ZFS:

- O conjunto de armazenamento do ZFS detalhado pode ser exibido através da utilização do comando `zpool status`.
- As falhas do conjunto e do dispositivo são relatadas através de mensagens de diagnóstico ZFS/FMA.
- Os comandos do ZFS anteriores que modificaram as informações de estado do conjunto podem ser exibidos através da utilização do comando `zpool history`.

A maioria das soluções de problemas do ZFS envolvem o comando `zpool status`. Esse comando analisa as diferentes falhas no sistema e identifica o problema mais grave, apresentando-lhe ações sugeridas e um link a um artigo com mais informações. Observe que o comando identifica somente um único problema com o conjunto, embora existam vários problemas. Por exemplo, erros de dados corrompidos geralmente implicam que um dos dispositivos falhou, mas substituir os dispositivos falhos não deve resolver todos os problemas de corrupção de dados.

Além disso, um diagnóstico do ZFS gera diagnósticos e relata falhas de conjuntos e de dispositivos. Soma de verificação, E/S, dispositivo, e erros de conjuntos associados com essas falhas também são reportadas. As falhas do ZFS, conforme relatadas por `fmd`, são exibidas no console, bem como no arquivo de mensagens do sistema. Na maioria dos casos, a mensagem `fmd` leva você ao comando `zpool status` para obter instruções de recuperação adicionais.

O processo básico de recuperação realiza-se da seguinte forma:

- Se for apropriado, utilize o comando `zpool history` para identificar os comandos do ZFS que precedem cenário do erro. Por exemplo:

```
# zpool history tank
History for 'tank':
2010-07-15.12:06:50 zpool create tank mirror c0t1d0 c0t2d0 c0t3d0
2010-07-15.12:06:58 zfs create tank/erick
2010-07-15.12:07:01 zfs set checksum=off tank/erick
```

Nessa saída, note que as somas de verificação estão desativadas no sistema de arquivos `tank/erick`. Esta configuração não é recomendável.

- Identifique os erros através das mensagens `fmd` exibidas no console do sistema ou nos arquivos `/var/adm/messages`.
- Localize as instruções de reparação adicionais através da utilização do comando `zpool status -x`.
- Reparar falhas envolve as etapas a seguir:
 - Substitua o dispositivo defeituoso ou ausente e o coloque on-line.
 - Restaure a configuração defeituosa ou os dados corrompidos a partir de um backup.
 - Verifique a recuperação utilizando o comando `zpool status -x`.
 - Efetue backup da configuração restaurada, se aplicável.

Essa seção descreve como interpretar a saída de `zpool status` a fim de diagnosticar os tipos de falhas que podem ocorrer. Embora a maioria do trabalho seja efetuado automaticamente pelo comando, é importante entender exatamente que problemas estão sendo identificados a fim de diagnosticar o tipo da falha. Seções subsequentes descrevem como reparar vários problemas que você pode encontrar.

Determinando se há problemas em um conjunto de armazenamento do ZFS

A forma mais fácil de determinar se há problemas conhecidos no sistema é utilizar o comando `zpool status -x`. Esse comando descreve somente os conjuntos que apresentam problemas. Se não há nenhum conjunto defeituoso no sistema, então o comando exibe o seguinte:

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

Sem o sinalizador `-x`, o comando exibe o status completo de todos os pools (ou do pool solicitado, se especificado na linha de comando), mesmo que os pools não apresentem falhas.

Para obter mais informações sobre opções de linha de comando para o comando `zpool status`, consulte [“Consultando status de pool de armazenamento do ZFS”](#) na página 106.

Revisando a saída de `zpool status`

A saída de `zpool status` completa se assemelha ao ilustrado abaixo:

```
# zpool status tank
# zpool status tank
  pool: tank
  state: DEGRADED
  status: One or more devices could not be opened. Sufficient replicas exist for
         the pool to continue functioning in a degraded state.
  action: Attach the missing device and online it using 'zpool online'.
         see: http://www.sun.com/msg/ZFS-8000-2Q
  scrub: none requested
  config:
```

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

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

Essa saída é descrita a seguir:

Informações gerais sobre o status do pool

Essa seção na saída de `zpool status` contém os campos a seguir, alguns dos quais são exibidos somente para os conjuntos que apresentam problemas:

<code>pool</code>	Identifique o nome do conjunto.
<code>state</code>	Indique a integridade atual do conjunto. Esta informação se refere somente a capacidade que o pool apresenta de oferecer o nível de replicação necessário.
<code>status</code>	Descreva o que há de errado com o conjunto. Esse campo é omitido se nenhum erro for encontrado.
<code>action</code>	Uma ação recomendada para a reparação de erros. Esse campo é omitido se nenhum erro for encontrado.
<code>see</code>	Referência a um artigo conhecido com informações sobre a reparação. Os artigos on-line são atualizados com mais frequência do que guias podem ser atualizados. Então, sempre os referencie para o mais atualizado procedimento de reparação. Esse campo é omitido se nenhum erro for encontrado.

- scrub Identifique o status atual da operação de scrub, que deve incluir a data e a hora de conclusão do último scrub realizado, um scrub em andamento ou se nenhum scrub foi solicitado.
- errors Identifica erros de dados ou ausência de erros de dados conhecidos.

Informações sobre a configuração do pool

O campo `config` na saída `zpool status` descreve a configuração dos dispositivos no conjunto, bem como seu estado e quaisquer erros gerados a partir dos dispositivos. O estado pode ser um dos seguintes: `ONLINE`, `FAULTED`, `DEGRADED`, `UNAVAIL` ou `OFFLINE`. Se for exibido somente `ONLINE`, a tolerância a falhas do pool foi comprometida.

A segunda seção da saída de configuração exibe os erros de estatísticas. Estes erros estão divididos em três categorias:

- `READ`: erros de E/S ocorridos durante uma solicitação de leitura
- `WRITE`: erros de E/S ocorridos durante uma solicitação de gravação
- `CKSUM`: erros de soma de verificação, significam que o dispositivo retornou dados corrompidos como resultado de uma requisição de leitura

Estes erros podem ser usados para determinar se o dano é permanente. Uma pequena quantidade de erros de E/S pode indicar uma interrupção temporária, enquanto que uma grande quantidade pode indicar um problema permanente com o dispositivo. Estes erros não correspondem necessariamente à corrupção de dados conforme interpretado pelos aplicativos. Se o dispositivo estiver em uma configuração redundante, os dispositivos podem exibir erros incorrigíveis, enquanto nenhum erro aparece no espelho ou no nível do dispositivo RAID-Z. Em tais casos, o ZFS recupera com sucesso os dados bons e tenta reabilitar dados danificados a partir das réplicas existentes.

Para mais informações sobre a interpretação desses erros, consulte [“Determinando o tipo de falha do dispositivo” na página 301](#).

Finalmente, na última coluna da saída de `zpool status` se encontram informações adicionais auxiliares. Essas informações abrangem o campo `estado`, auxiliando em diagnósticos de falhas. Se um dispositivo apresenta o estado `FAULTED`, este campo indica se o dispositivo encontra-se inacessível ou se os dados do dispositivo estão corrompidos. Se o dispositivo estiver sendo submetido a um `resilvering`, este campo exibe o progresso atual.

Para mais informações sobre a monitoração do progresso de `resilvering`, consulte [“Exibindo o status do `resilvering`” na página 309](#).

Status do scrubbing

A terceira seção de scrub da saída `zpool status` descreve o status atual de quaisquer operações de scrubbing explícitas. Se qualquer tipo de erro for detectado no sistema, estas informações serão diferentes, embora possam ser usadas para determinar a precisão do relatório de erro de corrupção de dados. Se o último scrubbing acabou de ser concluído, provavelmente nenhuma corrupção de dados conhecida foi encontrada.

Mensagens de finalização de scrub persistem através da reinicialização do sistema.

Para mais informações sobre o scrubbing de dados e como interpretar essas informações, consulte [“Verificando a integridade do sistema de arquivos ZFS” na página 291](#).

Erros de corrupção de dados

O comando `zpool status` mostra também se os erros conhecidos estão associados ao pool. Esses erros podem ter sido encontrados durante o scrubbing de dados ou durante uma operação normal. O ZFS mantém um log persistente de todos os erros de dados associados ao conjunto. Este registro é alternado sempre que um scrubbing completo do sistema for concluído.

Os erros de corrupção de dados são sempre fatais. A presença de tais erros indica que como mínimo um aplicativo sofreu um erro de E/S devido a dados corrompidos dentro do pool. Os erros de dispositivo dentro de um pool redundante não têm como resultado a corrupção de dados e não são registrados como parte deste registro. Por padrão, somente o número de erros encontrado é exibido. Uma lista completa de erros e de suas condições específicas pode ser encontrada usando a opção `-v` do `zpool status`. Por exemplo:

```
# zpool status -v
pool: tank
state: UNAVAIL
status: One or more devices are faulted in response to IO failures.
action: Make sure the affected devices are connected, then run 'zpool clear'.
       see: http://www.sun.com/msg/ZFS-8000-HC
scrub: scrub completed after 0h0m with 0 errors on Tue Feb  2 13:08:42 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM	
tank	UNAVAIL	0	0	0	insufficient replicas
clt0d0	ONLINE	0	0	0	
clt1d0	UNAVAIL	4	1	0	cannot open

```
errors: Permanent errors have been detected in the following files:
```

```
/tank/data/aaa
/tank/data/bbb
/tank/data/ccc
```

Uma mensagem semelhante também é exibida pelo `cmd` no console do sistema e no arquivo `/var/adm/messages`. Estas mensagens também podem ser rastreadas através do comando `cmdump`.

Para obter mais informações sobre a interpretação dos erros de corrupção de dados, consulte [“Identificando o tipo de corrupção de dados” na página 311](#).

Relatório de mensagens de erros do ZFS do sistema

Além de manter um rastreamento persistente de erros dentro do conjunto, o ZFS exibe também mensagens `syslog` quando ocorrem eventos de interesse. As situações abaixo geram eventos para notificar o administrador:

- **Transição de estado do dispositivo** – Se um dispositivo passa a ser `FAULTED`, o ZFS registra uma mensagem indicando que a tolerância a falhas do pool pode estar em risco. Uma mensagem semelhante é enviada se o dispositivo passar a ser `on-line`, restaurando a normalidade do pool.
- **Corrupção de dados** – Se for detectado qualquer tipo de corrupção de dados, o ZFS registra uma mensagem descrevendo quando e onde a corrupção foi detectada. Esta mensagem é registrada somente na primeira vez que a corrupção é detectada. Os acessos subsequentes não geram mensagens.
- **Falhas de conjunto e de dispositivos**: se ocorrer uma falha de conjunto ou de dispositivo, o daemon do gerenciador de falhas relata esses erros através de mensagens `syslog` bem como o `cmdump`.

Se o ZFS detectar um erro de dispositivo e se recuperar automaticamente, não ocorre nenhuma notificação. Tais erros não representam uma falha na integridade dos dados ou de redundância do conjunto. São normalmente o resultado de um problema do driver acompanhado por seu próprio conjunto de mensagens de erro.

Reparando uma configuração do ZFS danificada

O ZFS mantém um cache de conjuntos ativos e de suas configurações no sistema de arquivos raiz. Se esse arquivo for corrompido ou, de alguma forma, perder a sincronia com a informações de configuração que está armazenado no disco, o conjunto não poderá mais ser aberto. O ZFS tenta evitar essa situação, embora sempre exista a possibilidade de haver corrupções arbitrárias dadas as qualidades do armazenamento subjacente. Esta situação resulta normalmente no desaparecimento de um pool do sistema quando tal pool deveria, na verdade, estar disponível. Essa situação também pode manifestar-se como uma configuração parcial com

ausência de um número desconhecido de dispositivos virtuais de nível superior. Em ambos os casos, a configuração pode ser recuperada exportando o conjunto (se estiver visível) e importando-o novamente.

Para mais informações sobre a importação e exportação de conjuntos, consulte [“Migrando pools de armazenamento do ZFS” na página 115](#).

Reparando um dispositivo faltando

Se um dispositivo não puder ser aberto, será exibido como UNAVAILABLE na saída `zpool status`. Esse status significa que o ZFS não pode abrir o dispositivo quando o conjunto foi acessado pela primeira vez ou que o dispositivo se tornou indisponível a partir desse momento. Se o dispositivo fizer com que o dispositivo virtual de nível superior não esteja disponível, então não será possível acessar nada no pool. Do contrário, a tolerância a falhas do pool pode ser comprometida. Em ambos os casos, o dispositivo precisa simplesmente ser reanexado ao sistema para recuperar a operação normal.

Por exemplo, uma mensagem do `zfs` semelhante à ilustrada abaixo deve ser vista depois de uma falha no dispositivo:

```
SUNW-MSG-ID: ZFS-8000-FD, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Thu Jun 24 10:42:36 PDT 2010
PLATFORM: SUNW,Sun-Fire-T200, CSN: -, HOSTNAME: neo2
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: a1fb66d0-cc51-cd14-a835-961c15696fcb
DESC: The number of I/O errors associated with a ZFS device exceeded
acceptable levels. Refer to http://sun.com/msg/ZFS-8000-FD for more information.
AUTO-RESPONSE: The device has been offlined and marked as faulted. An attempt
will be made to activate a hot spare if available.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.
```

Para visualizar mais informações detalhadas sobre problemas e resoluções de dispositivos, utilize o comando `zpool status -x`. Por exemplo:

```
# zpool status -x
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: scrub completed after 0h0m with 0 errors on Tue Feb 2 13:15:20 2010
config:
```

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

```
c1t1d0 UNAVAIL      0      0      0 cannot open
```

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

É possível observar nessa saída que o dispositivo `c1t1d0` ausente não está funcionando. Se você determinar que o dispositivo é falho, substitua-o.

Depois, utilize o comando `zpool online` para colocar o dispositivo substituído on-line. Por exemplo:

```
# zpool online tank c1t1d0
```

Como uma última etapa, confirme que o conjunto com o dispositivo substituído está íntegro. Por exemplo:

```
# zpool status -x tank
pool 'tank' is healthy
```

Reanexando fisicamente um dispositivo

A forma como é realizada a reanexação de um dispositivo ausente depende do dispositivo em questão. Se o dispositivo for um drive anexado à rede, a conectividade deve ser restaurada. Se o dispositivo for um dispositivo USB ou outro tipo de mídia removível, deve ser reanexado ao sistema. Se o dispositivo for um disco local, um controlador pode ter falhado, de forma que tal dispositivo não está mais visível no sistema. Nesse caso, o controlador deve ser substituído no momento em que os discos estarão novamente disponíveis. Podem existir outros problemas que dependem do tipo de hardware e de sua configuração. Se uma unidade falhar e não estiver mais visível no sistema, o dispositivo deve ser tratado como um dispositivo danificado. Siga os procedimentos em [“Substituindo ou reparando um dispositivo modificado”](#) na página 301.

Notificando o ZFS da disponibilidade de um dispositivo

Depois que um dispositivo é reanexado ao sistema, o ZFS pode ou não detectar automaticamente sua disponibilidade. Se o conjunto já apresentava defeito anteriormente ou se o sistema tiver sido reinicializado como parte do procedimento anexar, então o ZFS reexamina todos os dispositivos ao tentar abrir o conjunto. Se o conjunto estava danificado e o dispositivo foi substituído enquanto o sistema estava em execução, é necessário notificar ao ZFS que o dispositivo está disponível e pronto para ser reaberto utilizando o comando `zpool online`. Por exemplo:

```
# zpool online tank c0t1d0
```

Para obter mais informações sobre como colocar dispositivos on-line, consulte [“Colocando um dispositivo on-line”](#) na página 94.

Substituindo ou reparando um dispositivo modificado

Esta seção descreve como determinar os tipos de falha do dispositivo, limpar os erros transitórios e substituir um dispositivo.

Determinando o tipo de falha do dispositivo

O termo *dispositivo danificado* é um tanto vago e pode descrever vários tipos de situações possíveis:

- **Bit rot:** com o tempo, eventos aleatórios, como influências magnéticas e raios cósmicos, podem fazer com que os bits armazenados no disco se invertam. Estes eventos são relativamente raros, mas comuns o suficiente para provocar corrupção de dados em sistemas grandes ou que estão em funcionamento durante longos períodos de tempo.
- **Leituras ou gravações mal endereçadas** – Erros de firmware ou falhas de hardware podem fazer com que leituras e gravações de blocos inteiros façam referência a locais incorretos no disco. Esses erros são normalmente transitórios, embora uma grande quantidade pode indicar um drive defeituosa.
- **Erro do administrador:** os administradores podem substituir involuntariamente partes do disco por dados ruins (como copiar /dev/zero sobre partes do disco) que provocam a corrupção permanente deste. Estes erros são sempre transitórios.
- **Interrupções temporárias** – Um disco pode não estar disponível durante um período de tempo, causando falhas de E/S. Esta situação está associada geralmente a dispositivos anexados à rede, embora os discos locais também possam sofrer interrupções temporárias. Estes erros podem ou não ser transitórios.
- **Hardware defeituoso ou anormal:** essa situação é um resumo de todos os vários problemas que hardware defeituoso exibe, incluindo erros de E/S de consistência, transportes causando corrupção aleatória ou alguns números de falhas. Estes erros são normalmente permanentes.
- **Dispositivo off-line:** se um dispositivo estiver off-line, supõe-se que o administrador o colocou nesse estado porque estava defeituoso. O administrador que colocou o dispositivo nesse estado pode determinar se esta suposição é precisa.

A determinação exata do problema pode ser um processo difícil. A primeira etapa é examinar as contagens de erros na saída `zpool status`. Por exemplo:

```
# zpool status -v tpool
pool: tpool
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: scrub completed after 0h0m with 2 errors on Tue Jul 13 11:08:37 2010
config:

NAME          STATE      READ WRITE CKSUM
tpool         ONLINE    2     0     0
  ct1t1d0     ONLINE    2     0     0
  ct1t3d0     ONLINE    0     0     0
errors: Permanent errors have been detected in the following files:

    /tpool/words
```

Os erros estão divididos em erros de E/S e erros de soma de verificação, e ambos podem indicar o possível tipo de falha. As operações normais prevêm uma pequena quantidade de erros (apenas alguns erros em longos períodos de tempo). Se você estiver vendo uma grande quantidade de erros, então essa situação provavelmente indica uma falha completa do dispositivo ou iminente. No entanto, um erro de administrador pode também resultar em uma grande contagem de erros. Outra fonte de informações é o log do sistema `syslog`. Se o registro mostrar um grande número de mensagens de driver de Fibre Channel ou de SCSI, então essa situação provavelmente indica sérios problemas de hardware. Se não for gerada nenhuma mensagem no `syslog`, então o dano é provavelmente transiente.

O objetivo é responder à seguinte pergunta:

É provável que ocorra outro erro neste dispositivo?

Os erros que acontecem somente uma vez são considerados *transiente* e não indicam falhas potenciais. Os erros que são persistentes ou suficientemente graves para indicar possível falha de hardware são considerados *fatais*. A ação de determinar o tipo de erro não está dentro do âmbito de nenhum software automatizado disponível atualmente com ZFS e muitas ações devem ser realizadas manualmente por você, o administrador. Depois de determinar o erro, a ação apropriada pode ser realizada. Apague os erros transitórios ou substitua o dispositivo devido aos erros fatais. Estes procedimentos de reparação estão descritos nas próximas seções.

Mesmo que os erros de dispositivo sejam considerados transientes, eles ainda podem ter provocado erros de dados incorrigíveis dentro do conjunto. Estes erros requerem procedimentos de reparação especiais, mesmo se o dispositivo estiver em boas condições ou tiver sido reparado. Para mais informações sobre a reparação de erros dos dados, consulte [“Reparando dados danificados” na página 310](#).

Apagando erros transitórios

Se os erros de dispositivo são considerados transientes, nesse caso eles não são suscetíveis de afetar a integridade futura do dispositivo e podem ser seguramente eliminados para indicar que nenhum erro fatal ocorreu. Para apagar os contadores de erros dos dispositivos RAID-Z ou espelhados, use o comando `zpool clear`. Por exemplo:

```
# zpool clear tank c1t1d0
```

Essa sintaxe elimina os erros do dispositivo e elimina quaisquer contagens de erros associadas a este dispositivo.

Para eliminar todos os erros associados aos dispositivos virtuais em um conjunto e quaisquer contagens de erros de dados associadas ao conjunto, utilize sintaxe a seguir:

```
# zpool clear tank
```

Para mais informações sobre como eliminar erros do conjunto, consulte [“Limando erros de dispositivo de conjunto de armazenamento”](#) na página 95.

Substituindo um dispositivo em um pool de armazenamento do ZFS

Se o dano no dispositivo for permanente ou se houver grande possibilidade de que ocorram futuros danos permanentes, o dispositivo deve ser substituído. Depende da configuração se o dispositivo pode ou não ser substituído.

- [“Determinando se um dispositivo pode ser substituído”](#) na página 303
- [“Dispositivos que não podem ser substituídos”](#) na página 304
- [“Substituindo um dispositivo em um pool de armazenamento do ZFS”](#) na página 304
- [“Exibindo o status do resilvering”](#) na página 309

Determinando se um dispositivo pode ser substituído

Para que um dispositivo possa ser substituído, o pool deve estar no estado `ONLINE`. O dispositivo deve fazer parte de uma configuração redundante ou deve estar em boas condições (no estado `ONLINE`). Se o dispositivo faz parte de uma configuração redundante, devem existir réplicas suficientes para recuperar os dados bons. Se dois discos em um espelho de quatro lados estiverem defeituosos, então ambos discos podem ser substituídos porque há cópias em boas condições disponíveis. No entanto, se dois discos de um dispositivo virtual (`raidz1`) RAID-Z quadridirecional estiverem defeituosos, então nenhum dos discos pode ser substituído porque não há cópias suficientes para recuperar os dados. Se o dispositivo estiver danificado, mas estiver on-line, poderá ser substituído, contanto que o pool não se encontre no estado `FAULTED`.

No entanto, os dados corrompidos do dispositivo são copiados no novo dispositivo, a menos que existam réplicas suficientes com dados bons.

Na configuração a seguir, o disco `c1t1d0` pode ser substituído e os dados do conjunto são copiados da réplica integral `c1t0d0`:

```
mirror          DEGRADED
c1t0d0          ONLINE
c1t1d0          FAULTED
```

O disco `c1t0d0` também pode ser substituído, embora a autocorreção dos dados não seja possível devido à falta de réplicas boas disponíveis.

Na configuração a seguir, nenhum dos discos defeituosos pode ser substituído. Os discos `ONLINE` também não podem ser substituídos porque o próprio conjunto está defeituoso.

```
raidz          FAULTED
c1t0d0          ONLINE
c2t0d0          FAULTED
c3t0d0          FAULTED
c4t0d0          ONLINE
```

Na configuração abaixo, ambos os discos de nível superior podem ser substituídos, embora os dados defeituosos presentes no disco sejam copiados no novo disco.

```
c1t0d0          ONLINE
c1t1d0          ONLINE
```

Se ambos os discos estiverem defeituosos, nenhuma substituição poderá ser efetuada porque o próprio conjunto pode estar defeituoso.

Dispositivos que não podem ser substituídos

Se a perda de um dispositivo tornar o conjunto defeituoso ou se o dispositivo contiver muitos erros de dados em uma configuração não redundante, então o dispositivo não poderá ser substituído com segurança. Sem redundâncias suficientes, não haverá dados bons com os quais reparar o dispositivo danificado. Neste caso, a única opção é destruir o conjunto e recriar a configuração e, em seguida, restaurar os dados de uma cópia de backup.

Para obter mais informações sobre a restauração de um pool inteiro, consulte [“Reparando o dano de todo o pool de armazenamento do ZFS” na página 313](#).

Substituindo um dispositivo em um pool de armazenamento do ZFS

Depois de ter determinado qual dispositivo pode ser substituído, utilize o comando `zpool replace` para substituí-lo. Se você estiver substituindo o dispositivo danificado por um diferente, utilize uma sintaxe similar a seguinte:


```
# zpool replace tank c1t1d0 c2t0d0
```

Este comando migra os dados para o novo dispositivo a partir do dispositivo danificado ou de outros dispositivos do conjunto se este estiver em uma configuração redundante. Quando o comando estiver concluído, ele separa o dispositivo danificado da configuração, momento no qual o dispositivo pode ser removido do sistema. Se você já tiver removido o dispositivo e o tiver substituído por um dispositivo novo no mesmo local, use a forma simples de dispositivo do comando. Por exemplo:

```
# zpool replace tank c1t1d0
```

Este comando pega um disco não formatado, o formata adequadamente e, em seguida, começa a realizar o resilvering dos dados do restante da configuração.

Para obter mais informações sobre o comando `zpool replace`, consulte [“Substituindo dispositivos em um pool de armazenamento” na página 95](#).

EXEMPLO 11-1 Substituindo um dispositivo em um pool de armazenamento do ZFS

O exemplo a seguir mostra como substituir um dispositivo (`c1t3d0`) no conjunto de armazenamento espelhado `tank` em um sistema Sun Fire x4500 do Oracle. Para substituir o disco `c1t3d0` por um novo disco no mesmo local (`c1t3d0`), desconfigure o disco antes de substituí-lo. A etapa básica segue:

- Coloque o disco off-line (`c1t3d0`) para ser substituído. Não é possível desconfigurar um disco que esteja sendo usado.
- Utilize o comando `cfgadm` para identificar o disco (`c1t3d0`) para ser desconfigurado e desconfigure-o. O conjunto estará degradado com o disco off-line nessa configuração espelhada, mas continuará disponível.
- Substitua fisicamente o disco (`c1t3d0`). Certifique-se de que o LED Ready to Remove azul esteja iluminado antes de remover fisicamente o drive defeituoso.
- Reconfigure o disco (`c1t3d0`).
- Coloque o novo disco (`c1t3d0`) novamente on-line.
- Execute o comando `zpool replace` para substituir o disco (`c1t3d0`).

Observação – Se você definiu anteriormente o `autoreplace` da propriedade do conjunto para `on`, qualquer dispositivo novo, encontrado no mesmo local físico que um dispositivo que antes pertencia ao conjunto, será automaticamente formatado e substituído sem o uso do comando `zpool replace`. Este recurso pode não ser suportado em todos os hardwares.

EXEMPLO 11-1 Substituindo um dispositivo em um pool de armazenamento do ZFS (Continuação)

- Se um disco defeituoso for substituído automaticamente por um sobressalente, pode ser necessário desanexar o sobressalente depois que o disco defeituoso for substituído. Por exemplo, se `c2t4d0` ainda é um sobressalente ativo depois que o disco falho é substituído, desanexe-o.

```
# zpool detach tank c2t4d0
```

O exemplo a seguir percorre através das etapas para substituir um disco em um conjunto de armazenamento do 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
```

Observe que o `zpool` output anterior pode mostrar o disco novo e o antigo abaixo do cabeçalho *replacing*. Por exemplo:

```
replacing    DEGRADED    0    0    0
c1t3d0s0/o  FAULTED     0    0    0
c1t3d0      ONLINE      0    0    0
```

EXEMPLO 11-1 Substituindo um dispositivo em um pool de armazenamento do ZFS (Continuação)

Esse texto significa que o processo de substituição está em andamento e que disco novo está sendo resilvered.

Se você for substituir um disco (c1t3d0) por outro (c4t3d0), então será necessário apenas executar o comando `zpool replace`. Por exemplo:

```
# 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
```

Você pode precisar executar o comando `zpool status` várias vezes até que a substituição do disco seja concluída.

```
# 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

EXEMPLO 11-2 Substituindo um dispositivo de registro com falhas

O exemplo a seguir mostra como recuperar de um dispositivo de log falho (c0t5d0 no conjunto de armazenamento pool). A etapa básica segue:

- Reveja a saída `zpool status -x` e a mensagem de diagnóstico FMA, aqui descrita: <http://www.sun.com/msg/ZFS-8000-K4>
- Substitua fisicamente o dispositivo de registro com falha.
- Coloque o dispositivo de log on-line.
- Limpe a condição de erro do pool.

```
# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
       Waiting for administrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
       or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:
```

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

<Physically replace the failed log device>

```
# zpool online pool c0t5d0
# zpool clear pool
```

```
# zpool status -x
pool: pool
state: FAULTED
status: One or more of the intent logs could not be read.
       Waiting for administrator intervention to fix the faulted pool.
action: Either restore the affected device(s) and run 'zpool online',
       or ignore the intent log records by running 'zpool clear'.
scrub: none requested
config:
```

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
```

Exibindo o status do resilvering

O processo de substituição de um dispositivo pode demorar um longo período de tempo, dependendo do tamanho do dispositivo e da quantidade de dados do conjunto. O processo de mover os dados de um dispositivo a outro é conhecido como *resilvering* e pode ser monitorado com a utilização do comando `zpool status`.

Os sistemas de arquivos tradicionais realizam resilvering de dados no nível do bloco. O ZFS, por eliminar a estrutura em camadas artificiais do gerenciador de volumes, pode realizar resilvering de uma forma muito mais eficaz e controlada. A duas principais vantagens deste recurso são:

- O ZFS realiza resilvering somente da quantidade mínima de dados necessária. No caso de uma curta interrupção (como oposição a uma substituição completa do dispositivo), o disco inteiro pode ser resilvered em questão de minutos ou segundos. Quando todo o disco é substituído, o processo de resilvering leva o tempo proporcional à quantidade de dados usados no disco. Substituir um disco de 500 GB pode levar alguns segundos se o conjunto só possuir poucos gigabytes de espaço em disco utilizado.
- O resilvering é um processo seguro e que pode ser interrompido. Se o sistema perder energia ou for reinicializado, o processo de resilvering recomeça exatamente de onde parou, sem necessidade de nenhuma intervenção manual.

Para exibir o processo de resilvering, use o comando `zpool status`. Por exemplo:

```
# zpool status tank
pool: tank
state: DEGRADED
status: One or more devices is currently being resilvered. The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scrub: resilver in progress for 0h0m, 22.60% done, 0h1m to go
config:
  NAME                STATE          READ WRITE CKSUM
  tank                 DEGRADED      0     0     0
  mirror-0            DEGRADED      0     0     0
  replacing-0         DEGRADED      0     0     0
  c1t0d0               UNAVAIL       0     0     0  cannot open
  c2t0d0               ONLINE        0     0     0  85.0M resilvered
  c1t1d0               ONLINE        0     0     0

errors: No known data errors
```

Neste exemplo, o disco `c1t0d0` está sendo substituído pelo `c2t0d0`. Esse evento é observado na saída de status pela presença do dispositivo virtual substituição na configuração. Esse dispositivo não é real, não é possível criar um conjunto utilizando-o. O propósito desse dispositivo é exclusivamente exibir o progresso do resilvering e identificar qual dispositivo está sendo substituído.

Note que qualquer conjunto atualmente submetido ao resilvering é colocado em estado `ONLINE` ou `DEGRADED` porque o conjunto não pode fornecer o nível desejado de redundância até o

processo de resilvering estar completo. Apesar de a E/S estar sempre programada com uma prioridade menor do que a E/S solicitada pelo usuário, o resilvering é realizado o mais rápido possível para minimizar o impacto no sistema. Depois que o resilvering estiver completo, a configuração reverte para a nova e completa configuração. Por exemplo:

```
# 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
```

O conjunto está novamente ONLINE e o disco falho original (c1t0d0) foi removido da configuração.

Reparando dados danificados

As seções seguintes descrevem como identificar o tipo de corrupção de dados e como reparar os dados, se possível.

- [“Identificando o tipo de corrupção de dados” na página 311](#)
- [“Reparando arquivos ou diretórios corrompidos” na página 312](#)
- [“Reparando o dano de todo o pool de armazenamento do ZFS” na página 313](#)

O ZFS utiliza a soma de verificação, redundância e autocorreção de dados para minimizar o risco de corrupção de dados. Por outro lado, a corrupção de dados pode ocorrer se um conjunto não é redundante, se a corrupção ocorreu enquanto um conjunto foi degradado ou uma série de eventos conspirados para corromper cópias múltiplas de uma parte dos dados. Em relação à origem, o resultado é o mesmo: Os dados estão corrompidos e, portanto, não podem ser acessados. A ação tomada depende do tipo de dados que estão corrompidos e de seu valor relativo. Podem ser corrompidos dois tipos básicos de dados:

- **Metadados do pool** – O ZFS requer uma determinada quantidade de dados a ser processada para abrir um pool e acessar os conjuntos de dados. Se esses dados estão corrompidos, todo o conjunto ou parte do conjunto de dados se tornará indisponível.
- **Dados do objeto** – Neste caso, a corrupção está dentro de um arquivo ou um diretório específico. Este problema pode fazer com que uma parte do arquivo ou diretório não possa ser acessada ou que o objeto fique totalmente danificado.

Os dados são verificados durante operações normais bem como através de um scrubbing. Para informações sobre como verificar a integridade do conjunto de dados, consulte [“Verificando a integridade do sistema de arquivos ZFS” na página 291.](#)

Identificando o tipo de corrupção de dados

Por padrão, o comando `zpool status` mostra somente que ocorreu uma corrupção, mas não mostra onde esta corrupção ocorreu. Por exemplo:

```
# zpool status monkey
pool: monkey
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
monkey	ONLINE	8	0	0
c1t1d0	ONLINE	2	0	0
c2t5d0	ONLINE	6	0	0

errors: 8 data errors, use `'-v'` for a list

Cada erro indica somente que um erro ocorreu em um dado momento. Os erros não estão necessariamente presentes no sistema até este momento. Sob circunstâncias normais, esse é o caso. Certas interrupções podem resultar em corrupção de dados que são automaticamente reparados depois que a interrupção acaba. É realizado um scrubbing completo do pool a fim de examinar todos os blocos ativos no pool, assim o registro do erro é redefinido sempre que o scrubbing terminar. Se detectar que os erros já não estão presentes e não quiser esperar a conclusão do scrubbing, redefina todos os erros no pool com o comando `zpool online`.

Se a corrupção de dados estiver nos metadados de todo o pool, a saída é um pouco diferente. Por exemplo:

```
# zpool status -v morpheus
pool: morpheus
id: 1422736890544688191
state: FAULTED
status: The pool metadata is corrupted.
action: The pool cannot be imported due to damaged devices or data.
see: http://www.sun.com/msg/ZFS-8000-72
config:
```

morpheus	FAULTED	corrupted data
c1t10d0	ONLINE	

No caso de corrupção de conjunto extenso, o conjunto é colocado em estado FAULTED porque o conjunto não pode fornecer o nível requerido de redundância.

Reparando arquivos ou diretórios corrompidos

Se um arquivo ou diretório está corrompido, o sistema ainda pode funcionar, dependendo do tipo de corrupção. Qualquer dano é efetivamente irrecuperável se não há nenhuma cópia boa de dados no sistema. Se os dados estão disponíveis, é preciso restaurar os dados afetados do backup. Mesmo assim, é possível recuperar essa corrupção sem ter que restaurar todo o pool.

Se o dano for dentro de um bloco de dados do arquivo, então o arquivo pode ser seguramente removido, eliminando o erro do sistema. Utilize o comando `zpool status -v` para exibir uma lista de nomes de arquivos com erros persistentes. Por exemplo:

```
# zpool status -v
pool: monkey
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: scrub completed after 0h0m with 8 errors on Tue Jul 13 13:17:32 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
monkey	ONLINE	8	0	0
c1t1d0	ONLINE	2	0	0
c2t5d0	ONLINE	6	0	0

```
errors: Permanent errors have been detected in the following files:
```

```
/monkey/a.txt
/monkey/bananas/b.txt
/monkey/sub/dir/d.txt
monkey/ghost/e.txt
/monkey/ghost/boo/f.txt
```

A lista de nomes de arquivos com erros persistentes devem ser descritos como os seguintes:

- Se o caminho completo do arquivo for encontrado e o conjunto de dados estiver montado, o caminho completo do arquivo é exibido. Por exemplo:

```
/monkey/a.txt
```

- Se o caminho completo do arquivo for encontrado, mas o conjunto de dados não estiver montado, então o nome do conjunto de dados é exibido sem a barra (/) precedente, seguido do caminho dentro do conjunto de dados do arquivo. Por exemplo:

```
monkey/ghost/e.txt
```


- Se o número do objeto não puder ser traduzido com sucesso para o caminho do arquivo, devido a um erro ou a que o objeto não possui um caminho de arquivo verdadeiro associado a ele, como é o caso de `dnode_t`, então o nome do conjunto de dados seguido do número do objeto é exibido. Por exemplo:

```
monkey/dnode: <0x0>
```

- Se um objeto no metaobject set (MOS) está corrompido, então uma marcação especial de `<metadata>`, seguida pelo número do objeto, é exibida.

Se a corrupção estiver dentro dos metadados de um arquivo ou diretório, a única opção é mover os arquivos para outro local. É possível mover seguramente qualquer arquivo ou diretório para um local menos conveniente, permitindo que o objeto original seja restaurado em seu lugar.

Reparando o dano de todo o pool de armazenamento do ZFS

Se o dano está nos metadados do conjunto e o dano previne o conjunto de ser aberto ou importado, então as opções a seguir estão disponíveis:

- Tente recuperar o conjunto utilizando o comando `zpool clear -F` ou o comando `zpool import -F`. Esses comandos tentam retornar as poucas últimas transações do conjunto para o estado operacional. É possível utilizar o comando `zpool status` para rever um conjunto danificado e as etapas de recuperação recomendadas. Por exemplo:

```
# zpool status
pool: tpool
state: FAULTED
status: The pool metadata is corrupted and the pool cannot be opened.
action: Recovery is possible, but will result in some data loss.
        Returning the pool to its state as of Wed Jul 14 11:44:10 2010
        should correct the problem. Approximately 5 seconds of data
        must be discarded, irreversibly. Recovery can be attempted
        by executing 'zpool clear -F tpool'. A scrub of the pool
        is strongly recommended after recovery.
        see: http://www.sun.com/msg/ZFS-8000-72
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM	
tpool	FAULTED	0	0	1	corrupted data
c1t1d0	ONLINE	0	0	2	
c1t3d0	ONLINE	0	0	4	

O processo de recuperação, como descrito acima, é utilizado pelo comando a seguir:

```
# zpool clear -F tpool
```

Se você tentar importar um conjunto de armazenamento danificado, irá visualizar mensagens similares as seguintes:

```
# zpool import tpool
cannot import 'tpool': I/O error
Recovery is possible, but will result in some data loss.
Returning the pool to its state as of Wed Jul 14 11:44:10 2010
should correct the problem. Approximately 5 seconds of data
must be discarded, irreversibly. Recovery can be attempted
by executing 'zpool import -F tpool'. A scrub of the pool
is strongly recommended after recovery.
```

O processo de recuperação, como descrito acima, é utilizado pelo comando a seguir:

```
# zpool import -F tpool
Pool tpool returned to its state as of Wed Jul 14 11:44:10 2010.
Discarded approximately 5 seconds of transactions
```

Se o conjunto danificado está no arquivo `zpool cache`, o problema é descoberto quando o sistema é inicializado e o conjunto danificado é reportado no comando `zpool status`. Se o conjunto não estiver no arquivo `zpool cache`, ele não irá importar nem abrir com sucesso e você verá mensagens de conjuntos danificados quando tentar importar o conjunto.

- Se o conjunto não pode ser recuperado pelo método de recuperação do conjunto descrito acima, restaure o conjunto e todos os seus dados pela cópia de backup. O mecanismo utilizado varia amplamente dependendo da configuração do conjunto e da estratégia de backup. Primeiro, salve a configuração como exibida pelo comando `zpool status`, assim é possível recriá-la depois que o conjunto for destruído. Então, utilize o comando `zpool destroy -f` para destruir o conjunto. Mantenha também, em um lugar seguro, um arquivo que descreva o layout dos conjuntos de dados e as diversas propriedades de conjunto, já que estas informações não poderão ser acessadas se o pool não puder ser acessado. Com a configuração do pool e o layout do conjunto de dados, você pode reconstruir a configuração completa depois de destruir o pool. Os dados podem, então, ser preenchidos usando o backup ou a estratégia de restauração de sua preferência.

Reparando um sistema não inicializável

O ZFS foi desenvolvido para ser robusto e estável apesar dos erros. Mesmo assim, erros de software ou certos problemas inesperados podem causar pane no sistema quando um conjunto é acessado. Como parte do processo de inicialização, cada pool deve ser aberto, o que significa que tais falhas farão com que o sistema entre em um ciclo de reinicialização por pane. Para recuperar a partir dessa situação, o ZFS deve ser informado para não procurar por qualquer conjunto ao iniciar.

O ZFS mantém um cache interno de pools disponíveis e suas configurações em `/etc/zfs/zpool.cache`. O local e o conteúdo deste arquivo são privados e estão sujeitos a alterações. Se o sistema torna-se não inicializável, inicialize as etapas none utilizando a opção de inicialização `-m milestone=one`. Depois que o sistema estiver inicializado, remonte o sistema de arquivos raiz como gravável e então renomeie ou mova o arquivo `/etc/zfs/zpool.cache` para outros locais. Essas ações fazem o ZFS esquecer que há quaisquer conjuntos existentes no

sistema, prevenindo-o de tentar acessar o conjunto defeituoso causando o problema. Você pode, então, passar para o estado normal do sistema usando o comando `svcadm mi lestone all`. É possível usar um processo semelhante ao inicializar a partir de uma raiz alternativa para realizar reparações.

Depois que o sistema estiver inicializado, tente importar o conjunto utilizando o comando `zpool import`. Entretanto, esta ação provavelmente provocará o mesmo erro ocorrido durante a inicialização, pois o comando usa o mesmo mecanismo para acessar os pools. Se houver vários pools no sistema, adote o seguinte procedimento:

- Renomeie ou mova o arquivo `zpool.cache` para outro local como discutido no texto anterior.
- Determine qual conjunto possui problemas utilizando o comando `fmddump -eV` para exibir os conjuntos com erros fatais relatados.
- Importe o conjunto um a um, ignorando os conjuntos que estão tendo problemas, como descrito na saída `fmddump`.

Descrição da versão do ZFS do Oracle Solaris

Este apêndice descreve as versões disponíveis do ZFS, os recursos de cada versão e Solaris OS que oferece o recurso e a versão do ZFS.

As seções a seguir são fornecidas neste apêndice:

- [“Visão geral das versões do ZFS” na página 317](#)
- [“Versões do conjunto do ZFS” na página 317](#)
- [“Versões do sistema de arquivos do ZFS” na página 318](#)

Visão geral das versões do ZFS

Novos recursos do sistema de arquivos e do conjunto do ZFS são introduzidos e podem ser acessados utilizando uma versão do ZFS específica, disponível nas versões do Solaris. É possível utilizar o comando `zpool upgrade` ou `zfs upgrade` para identificar se um conjunto ou sistema de arquivos se localiza em uma versão anterior que a versão atual do Solaris sendo executada oferece. Também é possível utilizar este comandos para atualizar as versões do conjunto e do sistema de arquivos.

Para obter informações sobre a utilização dos comandos `zpool upgrade` e `zfs upgrade`, consulte [“Atualizando os sistemas de arquivos do ZFS \(zfs upgrade\)” na página 33](#) e [“Atualizando pools de armazenamento do ZFS” na página 121](#).

Versões do conjunto do ZFS

A tabela a seguir fornece uma lista de versões do conjunto do ZFS disponíveis nas versões do Solaris.

Versão	Solaris 10	Descrição
1	Solaris 6 10/06	Initial ZFS version
2	Solaris 11 10/06	Ditto blocks (replicated metadata)
3	Solaris 11 10/06	Hot spares and double parity RAID-Z
4	Solaris 8 10/07	zpool history
5	Solaris 10 10/08	Algoritmo de compressão gzip
6	Solaris 10 10/08	Propriedade de conjunto bootfs
7	Solaris 10 10/08	Dispositivos de log intencional separados
8	Solaris 10 10/08	Administração delegada
9	Solaris 10 10/08	Propriedades refquota e refreservation
10	Solaris 10 5/09	Dispositivos de cache
11	Solaris 10 10/09	Melhor desempenho de limpeza
12	Solaris 10 10/09	Propriedades do instantâneo
13	Solaris 10 10/09	Propriedade snapused
14	Solaris 10 10/09	Propriedade aclinherit passthrough-x
15	Solaris 10 10/09	contabilidade de espaço de grupo e usuário
16	Solaris 10 9/10	Suporte para propriedades stmf
17	Solaris 10 9/10	Tripla paridade do RAID-Z
18	Solaris 10 9/10	Contenções do usuário do instantâneo
19	Solaris 10 9/10	Remoção do dispositivo de log
20	Solaris 10 9/10	Compression using zle (codificação de extensão zero)
21	Solaris 10 9/10	Reservado
22	Solaris 10 9/10	Propriedades recebidas

Versões do sistema de arquivos do ZFS

A tabela a seguir fornece uma lista de versões do sistema de arquivos do ZFS disponíveis nas versões do Solaris.

Versão	Solaris 10	Descrição
1	Solaris 6 10/06	Versão do sistema de arquivos do ZFS inicial

Versão	Solaris 10	Descrição
2	Solaris 10 10/08	Entradas do diretório aprimoradas
3	Solaris 10 10/08	Identificador exclusivo de sistemas de arquivos e diferenciação de maiúscula/minúscula (FIUD)
4	Solaris 10 10/09	Propriedades userquota e groupquota

Índice

A

acessando

instantâneo do ZFS
(exemplo de), 226

aclmode propriedade, 243

ACLs

ACL no arquivo do ZFS
descrição detalhada, 244

ACL no diretório do ZFS
descrição detalhada, 245

configurando ACLs no arquivo do ZFS (formato
verboso)
descrição, 246

definindo ACLs em arquivo do ZFS (formato
compacto)
(exemplo de), 259

definindo ACLs no arquivo do ZFS (formato
compacto)
descrição, 258

definindo em arquivos ZFS
descrição, 243

definindo herança de ACL no arquivo do ZFS
(formato verboso)
(exemplo de), 251

descrição, 237

descrição do formato, 239

diferenças das ACLs com base no rascunho
POSIX, 238

herança da ACL, 242

modificando a ACL comum no arquivo ZFS
(formato verboso)
(exemplo de), 247

ACLs (*Continuação*)

privilégios de acesso, 240

propriedade aclinherit, 242

propriedade aclmode, 243

propriedade da ACL, 242

restaurando ACL comum no arquivo do ZFS
(formato verboso)
(exemplo de), 250

sinalizadores de herança da ACL, 242

tipos de entrada, 240

ACLs com base no rascunho POSIX, descrição, 238

ACLs do NFSv4

descrição do formato, 239

ACLs do NFSv4, diferenças das ACLs com base no
rascunho POSIX, 238

ACLs do NFSv4

herança da ACL, 242

modelo

descrição, 237

propriedade da ACL, 242

sinalizadores de herança da ACL, 242

ACLs do Solaris

descrição do formato, 239

diferenças das ACLs com base no rascunho
POSIX, 238

herança da ACL, 242

novo modelo

descrição, 237

propriedade da ACL, 242

sinalizadores de herança da ACL, 242

adicionando

- discos a uma configuração RAID-Z (exemplo de), 84
- dispositivos a um conjunto de armazenamento do ZFS (zpool add) (exemplo de), 82
- dispositivos de cache (exemplo de), 86
- sistema de arquivos ZFS a uma zona não global (exemplo de), 281
- um dispositivo de log espelhado (exemplo de), 85
- volume ZFS a uma região não global (exemplo de), 282

administração delegada, visão geral, 265

administração delegada de ZFS, visão geral, 265

administração simplificada, descrição, 48

ajustando, tamanhos de dispositivos de permuta e despejo, 166

alocado propriedade, descrição, 104

altroot propriedade, descrição, 104

anexando

- dispositivos a um pool de armazenamento do ZFS (zpool attach) (exemplo de), 87

apagando

- erros de dispositivo (zpool clear) (exemplo de), 303

armazenamento em conjuntos, descrição, 45

arquivos, como componentes de pools de armazenamento do ZFS, 68

atime propriedade, descrição, 188

atualizando

- pool de armazenamento do ZFS descrição, 121

autocorreção de dados, descrição, 71

autoreplace propriedade, descrição, 104

B

blocos de inicialização, instalando com installboot e installgrub, 169

bootfs propriedade, descrição, 104

C

cache file propriedade, descrição, 104

canmount propriedade

descrição, 189

descrição detalhada, 198

clone, definição, 48

clones

criando (exemplo de), 228

destruindo (exemplo de), 229

recursos, 228

colocando dispositivos on-line e off-line

conjunto de armazenamento do ZFS

descrição, 92

colocando um dispositivo off-line (zpool offline)

conjunto de armazenamento do ZFS

(exemplo de), 93

colocando um dispositivo on-line

conjunto de armazenamento do ZFS (zpool online)

(exemplo de), 94

compartilhando

sistemas de arquivos ZFS

descrição, 213

exemplo de, 213

componentes de, pool de armazenamento do ZFS, 65

componentes do ZFS, requisitos para nomeação, 50

comportamento por espaço excedido, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 63

compressratio propriedade, descrição, 189

configuração espelhada

descrição, 69

recurso de redundância, 69

visão conceitual, 69

configuração RAID-Z

(exemplo de), 74

paridade dupla, descrição, 69

paridade única, descrição, 69

recurso de redundância, 69

visão conceitual, 69

configuração RAID-Z, adicionando discos a, (exemplo de), 84

configurando

ACLs no arquivo do ZFS file (formato verboso)

(descrição, 246

- configurando (*Continuação*)
 - cota do sistema de arquivos ZFS (`zfs set quota`)
 - exemplo de, 216
 - pontos de montagem ZFS (`zfs set mountpoint`)
 - (exemplo de), 210
 - ZFS at ime propriedade
 - (exemplo de), 203
- conjunto, definição, 49
- conjunto de armazenamento do ZFS
 - adicionando dispositivos a (`zpool add`)
 - (exemplo de), 82
 - dispositivo ausente (defeituoso)
 - descrição, 290
 - falhas, 289
 - processo de visualização de resilvering
 - (exemplo de), 309
 - versões
 - descrição, 317
- conjunto de dados
 - definição, 49
 - descrição, 184
- conjuntos de armazenamento do ZFS
 - colocando dispositivos on-line e off-line
 - descrição, 92
 - colocando um dispositivo off-line (`zpool offline`)
 - (exemplo de), 93
 - configuração espelhada, descrição, 69
 - configuração RAID-Z, descrição, 69
 - conjunto
 - definição, 49
 - criando (`zpool create`)
 - (exemplo de), 72
 - dados corrompidos
 - descrição, 290
 - determinando o tipo de falha do dispositivo
 - descrição, 301
 - determinando se um dispositivo pode ser substituído
 - descrição, 304
 - dispositivo virtual
 - definição, 50
 - dispositivos danificados
 - descrição, 290
 - distribuição dinâmica, 71
- conjuntos de armazenamento do ZFS (*Continuação*)
 - espelho
 - definição, 49
 - exibindo estado de integridade, 112
 - exibindo estado de integridade detalhado
 - (exemplo de), 114
 - exportando
 - (exemplo de), 116
 - informações gerais sobre o status do conjunto para
 - solução de problemas
 - descrição, 295
 - perfis de direitos, 287
 - RAID-Z
 - definição, 49
 - reparação de dados
 - descrição, 291
 - reparando uma configuração ZFS danificada, 299
 - resilvering
 - definição, 50
 - script de saída de conjunto de armazenamento
 - (exemplo de), 108
 - substituindo um dispositivo (`zpool replace`)
 - (exemplo de), 304
 - substituindo um dispositivo ausente
 - (exemplo de), 299
- conjuntos de armazenamento do ZFS (`zpool online`)
 - colocando um dispositivo on-line
 - (exemplo de), 94
- conjuntos de armazenamento ZFS
 - dividindo um conjunto de armazenamento
 - espelhado (`zpool split`)
 - (exemplo de), 89
 - exibindo estado de integridade
 - (exemplo de), 113
 - identificando para importar (`zpool import -a`)
 - (exemplo de), 117
 - ponto de montagem padrão, 81
 - reparando um sistema não inicializável
 - descrição, 314
 - resilvering e scrubbing de dados
 - descrição, 293
 - validação de dados
 - descrição, 292
- conjuntos de permissões, definidos, 265

- conjuntos raiz alternativo
 - criando
 - (exemplo de), 286
 - importando
 - (exemplo de), 287
 - contabilidade de espaço do ZFS, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 62
 - controlando, validação de dados (scrubbing), 292
 - cotas e reservas, descrição, 215
 - criação propriedade, descrição, 190
 - criando
 - clone do ZFS (exemplo de), 228
 - conjunto de armazenamento com dispositivos de cache (exemplo de), 76
 - conjunto de armazenamento de paridade dupla (zpool create)
 - (exemplo de), 74
 - conjunto de armazenamento do ZFS (zpool create)
 - (exemplo de), 72
 - conjunto de armazenamento RAID-Z de paridade tripla (zpool create)
 - (exemplo de), 74
 - conjuntos raiz alternativo
 - (exemplo de), 286
 - hierarquia do sistema de arquivos ZFS, 56
 - instantâneo do ZFS
 - (exemplo de), 222
 - pool de armazenamento do ZFS
 - descrição, 72
 - pool de armazenamento espelhado do ZFS (zpool create)
 - (exemplo de), 72
 - pool de armazenamento RAID-Z de paridade única (zpool create)
 - (exemplo de), 74
 - sistema de arquivos ZFS, 58
 - (exemplo de), 184
 - descrição, 184
 - um conjunto de armazenamento do ZFS (zpool create)
 - (exemplo de), 54
 - um conjunto de armazenamento ZFS com dispositivos de log (exemplo de), 75
 - criando (*Continuação*)
 - um novo conjunto ao dividir um conjunto de armazenamento espelhado (zpool split)
 - (exemplo de), 89
 - um sistema de arquivos do ZFS básico (zpool create)
 - (exemplo de), 54
 - volume do ZFS
 - (exemplo de), 277
- ## D
- dados
 - corrompidos, 290
 - corrupção identificada (zpool status -v)
 - (exemplo de), 297
 - reparação, 291
 - resilvering
 - descrição, 293
 - scrubbing
 - (exemplo de), 292
 - validação (scrubbing), 292
 - dados com soma de verificação, descrição, 47
 - definindo
 - ACLs em arquivo do ZFS (formato compacto)
 - (exemplo de), 259
 - ACLs em arquivos ZFS
 - descrição, 243
 - ACLs no arquivo do ZFS (formato compacto)
 - descrição, 258
 - cota do ZFS
 - (exemplo de), 204
 - herança de ACL no arquivo do ZFS (formato verboso)
 - (exemplo de), 251
 - propriedade `compression`
 - (exemplo de), 58
 - propriedade `mountpoint`, 58
 - propriedade `quota` (exemplo de), 58
 - propriedade `sharenfs`
 - (exemplo de), 58
 - reserva do sistema de arquivos ZFS
 - (exemplo de), 219

- delegando
 - conjunto de dados a uma região não global (exemplo de), 282
 - permissões (exemplo de), 270
- delegando permissões, `zfs allow`, 268
- delegando permissões a um grupo, (exemplo de), 270
- delegando permissões a um usuário individual, (exemplo de), 270
- desanexando
 - dispositivos de um pool de armazenamento do ZFS (`zpool detach`) (exemplo de), 89
- descompartilhando
 - sistemas de arquivos ZFS
 - exemplo de, 214
- desmontando
 - sistemas de arquivos ZFS (exemplo de), 212
- desmontar `zfs`, (exemplo de), 212
- despejo de memória, salvando, 168
- destruindo
 - clone do ZFS (exemplo de), 229
 - instantâneo do ZFS (exemplo de), 223
 - pool de armazenamento do ZFS
 - descrição, 72
 - pool de armazenamento do ZFS (`zpool destroy`) (exemplo de), 81
 - sistema de arquivos ZFS (exemplo de), 185
 - sistema de arquivos ZFS com dependentes (exemplo de), 186
- detectando
 - dispositivos em uso (exemplo de), 78
 - níveis de replicação inconsistentes (exemplo de), 80
- determinando
 - se um dispositivo pode ser substituído
 - descrição, 304
 - tipo de falha do dispositivo
 - descrição, 301
- diferenças entre o ZFS e os sistemas de arquivos tradicionais
 - diferenças entre o ZFS e os sistemas de arquivos tradicionais (*Continuação*)
 - comportamento por espaço excedido, 63
 - contabilidade de espaço do ZFS, 62
 - gerenciamento de volume tradicional, 63
 - granularidade do sistema de arquivos, 61
 - montando sistemas de arquivos ZFS, 63
 - novo modelo de ACL do Solaris, 63
 - discos, como componentes de pools de armazenamento do ZFS, 66
 - discos inteiros, como componentes de pools de armazenamento do ZFS, 66
 - disponível propriedade, descrição, 188
 - dispositivo virtual, definição, 50
 - dispositivos de cache
 - considerações para utilizar, 76
 - criando um conjunto de armazenamento ZFS com (exemplo de), 76
 - dispositivos de cache, adicionando, (exemplo de), 86
 - dispositivos de cache, removendo, (exemplo de), 86
 - dispositivos de intercâmbio e de dump, descrição, 165
 - dispositivos de log espelhados, criando um conjunto de armazenamento ZFS com (exemplo de), 75
 - dispositivos de log espelhados, adicionando, (exemplo de), 85
 - dispositivos de log separados, considerações sobre o uso, 34
 - dispositivos de permuta e de despejo, problemas, 165
 - dispositivos de permuta e despejo, ajustando tamanhos de, 166
 - dispositivos em uso
 - detectando (exemplo de), 78
 - dispositivos propriedade, descrição, 190
 - dispositivos virtuais, como componentes de pools de armazenamento do ZFS, 77
 - distribuição dinâmica
 - descrição, 71
 - recurso do conjunto de armazenamento, 71
 - dividindo um conjunto de armazenamento espelhado (`zpool split`) (exemplo de), 89
 - `dumpadm`, ativando um dispositivo de despejo, 168

E

- enviando e recebendo
 - dados do sistema de arquivos ZFS
 - descrição, 230
- espelho, definição, 49
- estado `zpool -x`, (exemplo de), 113
- exibindo
 - estado de integridade de conjunto de armazenamento ZFS (exemplo de), 113
 - estado de integridade detalhado do conjunto de armazenamento do ZFS (exemplo de), 114
 - estado de integridade dos conjuntos de armazenamento
 - descrição de, 112
 - estatísticas de E/S de pool de armazenamento do ZFS
 - descrição, 110
 - estatísticas de E/S de todos os pools de armazenamento do ZFS (exemplo de), 110
 - estatísticas de E/S de `vdev` do pool de armazenamento do ZFS (exemplo de), 111
 - histórico do comando, 39
 - permissões delegadas (exemplo de), 274
 - relatório de registro do sistema das mensagens de erro do ZFS
 - descrição, 298
- exportando
 - conjunto de armazenamento do ZFS (exemplo de), 116

F

- `failmode` propriedade, descrição, 105
- falhas, 289

G

- gerenciamento de volume tradicional, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 63

- granularidade do sistema de arquivos, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 61

H

- `health`, descrição, 105
- herdando
 - propriedades ZFS (`zfs inherit`)
 - descrição, 204
- hierarquia do sistema de arquivos, criando, 56
- histórico do comando, `zpool history`, 39

I

- identificando
 - conjunto de armazenamento ZFS para importar (`zpool import -a`) (exemplo de), 117
 - requisitos de armazenamento, 55
 - tipo de corrupção de dados `zpool status -v` (exemplo de), 311
- importando
 - conjuntos raiz alternativo (exemplo de), 287
 - pool de armazenamento do ZFS (exemplo de), 119
 - pool de armazenamento do ZFS a partir de diretórios alternativos (`zpool import -d`) (exemplo de), 118
- inicializando
 - sistema de arquivos raiz, 169
 - um BE do ZFS com `boot -L` e `boot -Z` em sistemas SPARC, 171
- instalação inicial do sistema de arquivos raiz ZFS, (exemplo de), 131
- instalação JumpStart
 - sistema de arquivos raiz
 - exemplos de perfil, 142
 - problemas, 142
- instalando
 - sistema de arquivo raiz ZFS
 - requisitos, 127

- instalando (*Continuação*)
 - sistema de arquivos raiz ZFS
 - (instalação inicial), 130
 - recursos, 126
 - sistema de arquivos ZFS
 - instalação JumpStart, 139
- instalando blocos de inicialização
 - `installboot` e `installgroup`
 - (exemplo de), 169
- instantâneo
 - acessando
 - (exemplo de), 226
 - contagem de espaço, 226
 - criando
 - (exemplo de), 222
 - definição, 50
 - destruindo
 - (exemplo de), 223
 - recursos, 221
 - renomeando
 - (exemplo de), 224
 - revertendo
 - (exemplo de), 227
- L**
- legenda EFI
 - descrição, 66
 - interação com o ZFS, 66
- limpando
 - um dispositivo em um pool de armazenamento do ZFS (`zpool clear`)
 - descrição, 95
- limpando um dispositivo
 - pool de armazenamento do ZFS)
 - (exemplo de), 95
- lista
 - informações do pool do ZFS, 56
 - pools de armazenamento do ZFS
 - descrição, 106
 - sistemas de arquivos do ZFS (`zfs list`)
 - (exemplo de), 59
- listando
 - descendentes dos sistemas de arquivos do ZFS
 - (exemplo de), 201
 - pools de armazenamento do ZFS
 - (exemplo de), 107
 - propriedades do ZFS (`zfs list`)
 - (exemplo de), 205
 - propriedades do ZFS por valor de origem
 - (exemplo de), 207
 - propriedades ZFS para scripting
 - (exemplo de), 207
 - sistemas de arquivos ZFS
 - (exemplo de), 201
 - sistemas de arquivos ZFS sem informações de cabeçalho
 - (exemplo de), 203
 - tipos de sistemas de arquivos ZFS
 - (exemplo de), 202
- `listsnapshots` propriedade, descrição, 105
- livre propriedade, descrição, 105
- log intencional do ZFS (ZIL), descrição, 34
- `luactivate`
 - sistema de arquivos raiz
 - (exemplo de), 147
- `lucreate`
 - BE do ZFS de um BE do ZFS
 - (exemplo de), 148
 - migração de sistema de arquivos raiz
 - (exemplo de), 145
- M**
- migrando
 - sistemas de arquivos raiz UFS para sistemas de arquivos raiz ZFS
 - (Oracle Solaris Live Upgrade), 143
 - problemas, 144
- migrando pools de armazenamento do ZFS,
 - descrição, 115
- modelo de ACL, Solaris, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 63
- modificando
 - ACL comum no arquivo do ZFS (formato verboso)
 - (exemplo de), 247

- modo de propriedade da ACL
 - acLinherit, 188
 - acLmode, 188
 - modos de falha
 - dados corrompidos, 290
 - dispositivo ausente (defeituoso), 290
 - dispositivos danificados, 290
 - montado propriedade, descrição, 190
 - montando
 - sistemas de arquivos ZFS (exemplo de), 211
 - montando sistemas de arquivos ZFS, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 63
- N**
- níveis de replicação inconsistentes
 - detectando (exemplo de), 80
 - notificando
 - ZFS sobre dispositivo reanexado (zpool onLine) (exemplo de), 300
- O**
- Oracle Solaris Live Upgrade
 - migração de sistema de arquivos raiz (exemplo de), 145
 - para migração de sistemas de arquivos raiz, 143
 - problemas de migração de sistemas de arquivos de raiz, 144
 - origem propriedade, descrição, 191
- P**
- palavras-chave de perfil do JumpStart, sistema de arquivos raiz ZFS, 140
 - perfis de direitos, para gerenciamento de sistemas de arquivos e conjuntos de armazenamento do ZFS, 287
 - ponto de montagem
 - padrão para conjuntos de armazenamento ZFS, 81
 - ponto de montagem (*Continuação*)
 - padrão para sistema de arquivos ZFS, 184
 - pontos de montagem
 - automático, 208
 - gerenciando ZFS
 - descrição, 209
 - legado, 209
 - pool de armazenamento espelhado (zpool create), (exemplo de), 72
 - pools de armazenamento do ZFS
 - anexando dispositivos a (zpool attach) (exemplo de), 87
 - apagando erros de dispositivo (zpool clear) (exemplo de), 303
 - atualizando
 - descrição, 121
 - componentes, 65
 - corrupção de dados identificada (zpool status -v) (exemplo de), 297
 - criando configuração espelhada (zpool create) (exemplo de), 72
 - criando uma configuração RAID-Z (zpool create) (exemplo de), 74
 - desanexando dispositivos de (zpool detach) (exemplo de), 89
 - destruindo (zpool destroy) (exemplo de), 81
 - determinando se há problemas (zpool status -x) descrição, 295
 - dispositivos virtuais, 77
 - estatísticas de E/S de vdev (exemplo de), 111
 - estatísticas de todos os pools (exemplo de), 110
 - identificando o tipo de corrupção de dados (zpool status -v) (exemplo de), 311
 - identificando problemas
 - descrição, 294
 - importando (exemplo de), 119
 - importando de diretórios alternativos (zpool import -d) (exemplo de), 118

- pools de armazenamento do ZFS (*Continuação*)
 - limpando um dispositivo
 - (exemplo de), 95
 - listando
 - (exemplo de), 107
 - mensagens de erro do sistema
 - descrição, 298
 - migrando
 - descrição, 115
 - notificando ZFS sobre dispositivo reanexado (zpool online)
 - (exemplo de), 300
 - pools de raiz alternativa, 286
 - realizando uma simulação (zpool create -n)
 - (exemplo de), 80
 - recuperando um pool destruído
 - (exemplo de), 120
 - reparando arquivos ou diretórios corrompidos
 - descrição, 312
 - reparando o dano de todo o pool
 - descrição, 314
 - scrubbing de dados
 - (exemplo de), 292
 - descrição, 292
 - substituindo um dispositivo (zpool replace)
 - (exemplo de), 95
 - usando arquivos, 68
 - usando discos inteiros, 66
- pools de raiz alternativa, descrição, 286
- primarycache propriedade, descrição, 191
- propriedade aclinherit, 242
- propriedade capacity, descrição, 104
- propriedade checksum, descrição, 189
- propriedade compression, descrição, 189
- propriedade configuráveis do ZFS, dispositivos, 190
- propriedade copies, descrição, 190
- propriedade de conjunto ZFS, cachefile, 104
- propriedade delegation, desativando, 266
- propriedade delegation, descrição, 105
- propriedade do pool ZFS, health, 105
- propriedade exec, descrição, 190
- propriedade guid, descrição, 105
- propriedade mountpoint, descrição, 190
- propriedade quota, descrição, 191
- propriedade sharenfs
 - descrição, 193, 213
- propriedade size, descrição, 106
- propriedade type, descrição, 194
- propriedade volsize, descrição, 195
- propriedade zoned
 - descrição, 195
 - descrição detalhada, 285
- propriedades configuráveis do ZFS
 - aclinherit, 188
 - aclmode, 188
 - atime, 188
 - canmount, 189
 - descrição detalhada, 198
 - descrição, 197
 - primarycache, 191
 - recordsize, 191
 - descrição detalhada, 199
 - refquota, 192
 - refreservation, 192
 - reserva, 193
 - secondarycache, 193
 - setuid, 193
 - shareiscsi, 193
 - snapdir, 194
 - somente leitura, 191
 - utilizadas
 - descrição detalhada, 196
 - versão, 195
 - volblocksize, 195
 - volsize
 - descrição detalhada, 199
 - xattr, 195
- propriedades de conjunto ZFS
 - failmode, 105
 - listsnapshots, 105
 - livre, 105
 - versão, 106
- propriedades de somente leitura do ZFS
 - compression, 189
 - criação, 190
 - descrição, 195
 - disponível, 188
 - montado, 190

- propriedades de somente leitura do ZFS (*Continuação*)
 - origem, 191
 - referenciado, 192
 - usedbychildren, 194
 - usedbydataset, 194
 - usedbyreservation, 194
 - utilizado, 194
- propriedades de ZFS, descrição de propriedades herdáveis, 187
- propriedades definíveis de ZFS
 - checksum, 189
 - volsize, 195
- propriedades definíveis do ZFS
 - compression, 189
 - copies, 190
 - exec, 190
 - mountpoint, 190
 - quota, 191
 - sharenfs, 193
 - zoned, 195
- propriedades do conjunto do ZFS
 - alroot, 104
 - bootfs, 104
- propriedades do conjunto ZFS
 - alocado, 104
 - autoreplace, 104
- propriedades do pool do ZFS
 - capacity, 104
 - delegation, 105
 - guid, 105
 - size, 106
- propriedades do usuário do ZFS
 - (exemplo de), 200
 - descrição detalhada, 200
- propriedades do ZFS
 - checksum, 189
 - compression, 189
 - configuráveis, 197
 - copies, 190
 - descrição, 187
 - exec, 190
 - gerenciamento dentro de uma região
 - descrição, 283
 - herdável, descrição de, 187
- propriedades do ZFS (*Continuação*)
 - mountpoint, 190
 - propriedade zoned
 - descrição detalhada, 285
 - quota, 191
 - sharenfs, 193
 - type, 194
 - volsize, 195
 - zoned, 195
- propriedades somente leitura do ZFS, type, 194
- propriedades ZFS
 - aclinherit, 188
 - aclmode, 188
 - atime, 188
 - available, 188
 - canmount, 189
 - descrição detalhada, 198
 - compressratio, 189
 - criação, 190
 - descrição, 187
 - dispositivos, 190
 - montado, 190
 - origem, 191
 - propriedades do usuário
 - descrição detalhada, 200
 - recordsize, 191
- Propriedades ZFS
 - recordsize
 - descrição detalhada, 199
- propriedades ZFS
 - referenciado, 192
 - refquota, 192
 - reservation, 192
 - reserva, 193
 - secondarycache, 191, 193
 - setuid, 193
 - shareiscsi, 193
- Propriedades ZFS
 - snapdir, 194
- propriedades ZFS
 - somente leitura, 191
 - somente leitura, 195
 - utilizadas
 - descrição detalhada, 196

propriedades ZFS (*Continuação*)

- utilizado, 194
- versão, 195
- volblocksize, 195

Propriedades ZFS

- volsize
 - descrição detalhada, 199

propriedades ZFS

- xattr, 195

R

RAID-Z, definição, 49

read-only properties of ZFS, usedbysnapshots, 194

recebendo

- dados do sistema de arquivos ZFS (zfs receive)
 - (exemplo de), 232

recordsize propriedade

- descrição, 191
- descrição detalhada, 199

recuperando

- pool de armazenamento destruído do ZFS
 - (exemplo de), 120

recursos de replicação de ZFS, espelhado ou

RAID-Z, 69

referenciado propriedade, descrição, 192

refquota propriedade, descrição, 192

refreservation propriedade, descrição, 192

regiões

- adicionando volume ZFS a uma região não global
 - (exemplo de), 282
- delegando conjunto de dados a uma região não global
 - (exemplo de), 282
- gerenciamento de propriedades do ZFS dentro de uma região
 - descrição, 283
- propriedade zoned
 - descrição detalhada, 285
- utilizando com sistemas de arquivos do ZFS
 - descrição, 280

removendo, dispositivos de cache (exemplo de), 86

removendo permissões, zfs unallow, 269

renomeando

- instantâneo do ZFS
 - (exemplo de), 224
- sistema de arquivos ZFS
 - (exemplo de), 186

reparando

- dano de todo o pool
 - descrição, 314
- reparando arquivos ou diretórios corrompidos
 - descrição, 312
- um sistema não inicializável
 - descrição, 314
- uma configuração ZFS danificada
 - descrição, 299

requisitos, para a instalação e o Live Upgrade, 127

requisitos de armazenamento, identificando, 55

requisitos de hardware e software, 53

requisitos para nomeação, componentes do ZFS, 50

reserva

- criando
 - (exemplo de), 97

reserva propriedade, descrição, 193

reservas

- descrição de
 - (exemplo de), 98

resilvering, definição, 50

resilvering e scrubbing de dados, descrição, 293

restaurando

- comum no arquivo do ZFS (formato verboso)
 - (exemplo de), 250

revertendo

- instantâneo do ZFS
 - (exemplo de), 227

S

salvando

- dados do sistema de arquivos do ZFS (zfs send)
 - (exemplo de), 231

desejos de memória

savecore, 168

savecore, salvando desejos de memória, 168

- script
 - saída de conjunto de armazenamento do ZFS (exemplo de), 108
- scrubbing
 - (exemplo de), 292
 - validação de dados, 292
- secondarycache propriedade, descrição, 193
- semânticas transacionais, descrição, 46
- setting
 - legacy mount points (exemplo de), 210
- setuid propriedade, descrição, 193
- shareiscsi propriedades, descrição, 193
- simulação
 - criação de pool de armazenamento do ZFS (zpool create -n) (exemplo de), 80
- sistema de arquivos, definição, 49
- sistema de arquivos do ZFS
 - criando um volume do ZFS (exemplo de), 277
 - versões
 - descrição, 317
- sistema de arquivos raiz ZFS
 - instalação inicial do sistema de arquivos raiz ZFS, 130
 - instalando um sistema de arquivos raiz, 126
- sistema de arquivos ZFS
 - descrição, 183
 - listando propriedades para scripting (exemplo de), 207
- sistemas de arquivo ZFS, requisitos de instalação e Live Upgrade, 127
- sistemas de arquivos do ZFS
 - ACL no arquivo do ZFS
 - descrição detalhada, 244
 - ACL no diretório do ZFS
 - descrição detalhada, 245
 - administração simplificada
 - descrição, 48
 - armazenamento em conjuntos
 - descrição, 45
 - clone, 228, 229
- sistemas de arquivos do ZFS (*Continuação*)
 - clones
 - definição, 48
 - configuração de ACLs no arquivo do ZFSe (formato verboso)
 - descrição, 246
 - conjunto de dados
 - definição, 49
 - contagem de espaço do instantâneo, 226
 - dados com soma de verificação
 - descrição, 47
 - definindo ACLs em arquivo do ZFS (formato compacto) (exemplo de), 259
 - definindo ACLs no arquivo do ZFS (formato compacto)
 - descrição, 258
 - definindo herança de ACL no arquivo do ZFS (formato verboso) (exemplo de), 251
 - delegando conjunto de dados a uma região não global (exemplo de), 282
 - descrição, 45
 - instantâneo
 - definição, 50
 - descrição, 221
 - modificando a ACL comum no arquivo do ZFS (formato verboso) (exemplo de), 247
 - perfis de direitos, 287
 - requisitos para nomeação de componentes, 50
 - restaurando ACL comum no arquivo do ZFS (formato verboso) (exemplo de), 250
 - salvando fluxos de dados (zfs send) (exemplo de), 231
 - semânticas transacionais
 - descrição, 46
 - sistema de arquivos
 - definição, 49
 - utilizando em um sistema Solaris com regiões instaladas
 - descrição, 280

- sistemas de arquivos do ZFS (*Continuação*)
 - volume
 - definição, 50
- sistemas de arquivos raiz ZFS
 - migração de sistemas de arquivos raiz com o Oracle Solaris Live Upgrade
 - (exemplo de), 145
- sistemas de arquivos ZFS
 - adicionando sistema de arquivos ZFS a uma zona não global
 - (exemplo de), 281
 - adicionando volume ZFS a uma região não global
 - (exemplo de), 282
 - clone
 - substituindo um sistema de arquivos por
 - (exemplo de), 229
 - clones
 - descrição, 228
 - compartilhando
 - descrição, 213
 - exemplo de, 213
 - configurando `atime` propriedade
 - (exemplo de), 203
 - criando
 - (exemplo de), 184
 - definindo a propriedade `quota`
 - (exemplo de), 204
 - definindo ACLs em arquivos ZFS
 - descrição, 243
 - definindo uma reserva
 - (exemplo de), 219
 - definir ponto de montagem (`zfs set mountpoint`)
 - (exemplo de), 210
 - descompartilhando
 - exemplo de, 214
 - desmontando
 - (exemplo de), 212
 - destruindo
 - (exemplo de), 185
 - destruindo com dependentes
 - (exemplo de), 186
 - dispositivos de intercâmbio e de dump
 - descrição, 165
- sistemas de arquivos ZFS (*Continuação*)
 - dispositivos de permuta e de despejo
 - problemas, 165
 - dispositivos de permuta e despejo
 - ajustando os tamanhos de, 166
 - enviando e recebendo
 - descrição, 230
 - gerenciamento dentro de uma região
 - descrição, 283
 - gerenciando pontos de montagem
 - descrição, 209
 - gerenciando pontos de montagem automáticos, 208
 - gerenciando pontos de montagem de legado
 - descrição, 209
 - herdando propriedade de (`zfs inherit`)
 - (exemplo de), 204
 - inicializando um BE do ZFS com `boot -L` e `boot -Z`
 - (exemplo de SPARC), 171
 - inicializando um sistema de arquivos raiz
 - descrição, 169
 - instalação JumpStart do sistema de arquivos raiz, 139
 - instantâneo
 - acessando, 226
 - criando, 222
 - renomeando, 224
 - revertendo, 227
 - instantâneos
 - destruindo, 223
 - listando
 - (exemplo de), 201
 - listando descendentes
 - (exemplo de), 201
 - listando propriedades de (`zfs list`)
 - (exemplo de), 205
 - listando propriedades por valor de origem
 - (exemplo de), 207
 - listando sem informações de cabeçalho
 - (exemplo de), 203
 - listando tipos de
 - (exemplo de), 202
 - migração de sistemas de arquivos raiz com o Oracle Solaris Live Upgrade, 143

- sistemas de arquivos ZFS (*Continuação*)
 - montando
 - (exemplo de), 211
 - ponto de montagem padrão
 - (exemplo de), 184
 - problemas de migração de sistemas de arquivos raiz, 144
 - recebendo fluxos de dados (zfs receive)
 - (exemplo de), 232
 - renomeando
 - (exemplo de), 186
 - setting legacy mount point
 - (exemplo de), 210
 - soma de verificação
 - definição, 48
 - tipos de conjunto de dados
 - descrição, 202
 - sistemas de arquivos ZFS (zfs set quota)
 - configurando uma cota
 - exemplo de, 216
 - snappdir propriedade, descrição, 194
 - solução de problemas
 - apagar erros de dispositivo(zpool clear)
 - (exemplo de), 303
 - corrupção de dados identificada (zpool status -v)
 - (exemplo de), 297
 - determinando o tipo de corrupção de dados (zpool status -v)
 - (exemplo de), 311
 - determinando o tipo de falha do dispositivo
 - descrição, 301
 - determinando se há problemas (zpool status -x), 295
 - determinando se um dispositivo pode ser substituído
 - descrição, 304
 - dispositivo ausente (defeituoso), 290
 - dispositivos danificados, 290
 - falhas do ZFS, 289
 - identificando problemas, 294
 - informações gerais sobre o status do conjunto
 - descrição, 295
 - notificando ZFS sobre dispositivo reanexado (zpool online)
 - (exemplo de), 300
 - solução de problemas (*Continuação*)
 - relatório de registro do sistema das mensagens de erro do ZFS, 298
 - reparando arquivos ou diretórios corrompidos
 - descrição, 312
 - reparando o dano de todo o pool
 - descrição, 314
 - reparando um sistema não inicializável
 - descrição, 314
 - reparando uma configuração ZFS danificada, 299
 - substituindo um dispositivo (zpool replace)
 - (exemplo de), 304, 309
 - substituindo um dispositivo ausente
 - (exemplo de), 299
 - soma de verificação, definição, 48
 - somente leitura propriedade, descrição, 191
 - substituindo
 - um dispositivo (zpool replace)
 - (exemplo de), 95, 304, 309
 - um dispositivo ausente
 - (exemplo de), 299
- ## T
- terminologia
 - clone, 48
 - conjunto, 49
 - conjunto de dados, 49
 - dispositivo virtual, 50
 - espelho, 49
 - instantâneo, 50
 - RAID-Z, 49
 - resilvering, 50
 - sistema de arquivos, 49
 - soma de verificação, 48
 - volume, 50
 - tipos de conjunto de dados, descrição, 202
- ## U
- usedbychildren propriedade, descrição, 194
 - usedbydataset propriedade, descrição, 194
 - usedbyreservation propriedade, descrição, 194

usedbysnapshots propriedade, descrição, 194
 utilizadas propriedades, descrição detalhada, 196
 utilizado propriedade, descrição, 194

V

verificar, integridade dos dados ZFS, 291
 versão do ZFS
 recurso do ZFS e Solaris
 descrição, 317
 versão propriedade, descrição, 195
 versão propriedade, descrição, 106
 volblocksize propriedade, descrição, 195
 volsize propriedade, descrição detalhada, 199
 volume, definição, 50
 volume do ZFS, descrição, 277

X

xattr propriedade, descrição, 195

Z

zfs allow
 descrição, 268
 exibindo permissões delegadas, 274
 zfs create
 (exemplo de), 58, 184
 descrição, 184
 zfs destroy, (exemplo de), 185
 zfs destroy -r, (exemplo de), 186
 zfs get, (exemplo de), 205
 zfs get -H -o, (exemplo de), 207
 zfs get -s, (exemplo de), 207
 zfs inherit, (exemplo de), 204
 zfs list
 (exemplo de), 59, 201
 zfs list -H, (exemplo de), 203
 zfs list -r, (exemplo de), 201
 zfs list -t, (exemplo de), 202
 zfs mount, (exemplo de), 211
 zfs promote, promoção de clone (exemplo de), 229

ZFS properties
 usedbychildren, 194
 usedbydataset, 194
 usedbyreservation, 194
 usedbysnapshots, 194
 zfs receive, (exemplo de), 232
 zfs rename, (exemplo de), 186
 zfs send, (exemplo de), 231
 zfs set atime, (exemplo de), 203
 zfs set compression, (exemplo de), 58
 zfs set mountpoint
 (exemplo de), 58, 210
 zfs set mountpoint=legacy, (exemplo de), 210
 zfs set quota
 (exemplo de), 58
 zfs set quota, (exemplo de), 204
 zfs set quota
 exemplo de, 216
 zfs set reservation, (exemplo de), 219
 zfs set sharenfs, (exemplo de), 58
 zfs set sharenfs=on, exemplo de, 213
 zfs unallow, descrição, 269
 zonas
 adicionando o sistema de arquivos ZFS a uma zona
 não global
 (exemplo de), 281
 zpool add, (exemplo de), 82
 zpool attach, (exemplo de), 87
 zpool clear
 (exemplo de), 95
 descrição, 95
 zpool create
 (exemplo de), 54, 56
 conjunto básico
 (exemplo de), 72
 pool de armazenamento espelhado
 (exemplo de), 72
 pool de armazenamento RAID-Z
 (exemplo de), 74
 zpool create -n, simulação (exemplo de), 80
 zpool destroy, (exemplo de), 81
 zpool detach, (exemplo de), 89
 zpool export, (exemplo de), 116
 zpool history, (exemplo de), 39

zpool import -a, (exemplo de), 117
zpool import -D, (exemplo de), 120
zpool import -d, (exemplo de), 118
zpool import *nome*, (exemplo de), 119
zpool iostat, todos os pools (exemplo de), 110
zpool iostat -v, vdev (exemplo de), 111
zpool list
 (exemplo de), 56, 107
 descrição, 106
zpool list -Ho name, (exemplo de), 108
zpool offline, (exemplo de), 93
zpool online, (exemplo de), 94
zpool replace, (exemplo de), 95
zpool split, (exemplo de), 89
zpool status -v, (exemplo de), 114
zpool upgrade, 121