

Guia de administração do Oracle® Solaris ZFS

Copyright © 2006, 2013, 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 END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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.

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. AMD, Opteron, o logotipo da AMD e o logotipo do AMD Opteron são marcas comerciais ou marcas comerciais registradas da Advanced Micro Devices. UNIX é uma marca comercial registrada licenciada por meio do consórcio The Open Group.

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.

Conteúdo

Prefácio	11
1 Sistema de arquivos do Oracle Solaris ZFS (introdução)	15
O que há de novo no ZFS?	15
Aprimoramentos de Uso do Comando ZFS	16
Aprimoramento de Instantâneo do ZFS	16
PropriedadeaclmodeAprimorada	17
Recursos de instalação do Oracle Solaris ZFS	17
Aprimoramentos no fluxo de envio do ZFS	18
Diferenças do instantâneo do ZFS (zfs diff)	18
Recuperação do pool de armazenamento do ZFS e aprimoramentos no desempenho	18
Ajuste do comportamento síncrono do ZFS	19
Mensagens aprimoradas do pool ZFS	19
Aprimoramentos de interoperabilidade ACL do ZFS	20
Dividindo um pool de armazenamento do ZFS espelhado (zpool split)	22
Novo processo do sistema ZFS	22
Aprimoramentos de substituição de dispositivo do ZFS	22
Suporte de instalação do ZFS e Flash	24
Migração de zonas em um ambiente do ZFS	24
Suporte à inicialização e instalação do ZFS	24
Gerenciamento do ZFS baseado na Web	24
O que é Oracle Solaris ZFS?	25
Armazenamento de ZFS em pool	25
Semânticas transacionais	26
Somos de verificação e autocorreção de dados	26
Escalabilidade não paralela	27
Instantâneos do ZFS	27
Administração simplificada	27

Terminologia do ZFS	28
Requisitos para nomeação de componentes do ZFS	30
Diferenças entre o sistema de arquivos tradicional e o Oracle Solaris ZFS	31
Granularidade do sistema de arquivos ZFS	31
Contabilidade de espaço em disco do ZFS	32
Montando sistemas de arquivos ZFS	34
Gerenciamento de volume tradicional	34
Modelo de ACL do Solaris Baseado em NFSv4	34
2 Introdução ao ZFS do Oracle Solaris	35
Perfis de direitos do ZFS	35
Recomendações e requisitos de software e hardware do ZFS	36
Criando um sistema de arquivos ZFS básico	36
Criando um pool de armazenamento básico do ZFS	37
▼ Como identificar os requisitos de armazenamento para o pool de armazenamento do ZFS	37
▼ Como criar um pool de armazenamento do ZFS	38
Criando uma hierarquia de sistemas de arquivos ZFS	39
▼ Como determinar a hierarquia do sistema de arquivos ZFS	39
▼ Como criar sistemas de arquivos ZFS	40
3 Gerenciando pools de armazenamento do Oracle Solaris ZFS	43
Componentes de um pool de armazenamento do ZFS	43
Usando discos em um pool de armazenamento do ZFS	43
Usando faixas em um pool de armazenamento do ZFS	44
Usando arquivos em um pool de armazenamento do ZFS	46
Considerações para Pools de Armazenamento do ZFS	46
Recursos de replicação de um pool de armazenamento do ZFS	47
Configuração de pool de armazenamento espelhado	47
Configuração de pool de armazenamento RAID-Z	47
Pool armazenamento híbrido do ZFS	49
Dados de autocorreção em uma configuração redundante	49
Distribuição dinâmica em um pool de armazenamento	49
Criando e destruindo pools de armazenamento do ZFS	50
Criando Pools de Armazenamento do ZFS	50

Exibindo informações do dispositivo virtual do pool de armazenamento	57
Manipulando erros de criação do pool de armazenamento do ZFS	58
Destruindo pools de armazenamento do ZFS	61
Gerenciando dispositivos em pools de armazenamento do ZFS	62
Adicionando dispositivos a um pool de armazenamento	62
Anexando e desanexando dispositivos em um pool de armazenamento	67
Criando um novo pool ao dividir um pool de armazenamento espelhado do ZFS	69
Colocando dispositivos off-line e on-line em um pool de armazenamento	72
Limpando erros de dispositivo do pool de armazenamento	74
Substituindo dispositivos em um pool de armazenamento	75
Designando discos de reserva com reposição em funcionamento no pool de armazenamento	78
Gerenciando propriedades do pool de armazenamento do ZFS	83
Consultando status de pool de armazenamento do ZFS	86
Exibindo informações sobre os pools de armazenamento do ZFS	86
Visualizando estatísticas de E/S para pools de armazenamento do ZFS	90
Determinando o status de integridade de pools de armazenamento do ZFS	92
Migrando pools de armazenamento do ZFS	97
Preparando para migração do pool de armazenamento do ZFS	98
Exportando um pool de armazenamento do ZFS	98
Determinando pools de armazenamento disponíveis para importação	99
Importando pools de armazenamento ZFS de diretórios alternativos	101
Importando pools de armazenamento do ZFS	101
Recuperando pools de armazenamento destruídos do ZFS	105
Atualizando pools de armazenamento do ZFS	106
4 Instalando e inicializando um sistema de arquivos raiz do Oracle Solaris ZFS	109
Instalando e inicializando um sistema de arquivos do Oracle Solaris ZFS (visão geral)	109
Recursos de instalação do ZFS	110
Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS	111
Instalação de um sistema de arquivos raiz ZFS (instalação inicial do Oracle Solaris)	114
▼ Como migrar um pool raiz espelhado do ZFS (pós-instalação)	120
Instalando um sistema de arquivos raiz ZFS (instalação de arquivo do Oracle Solaris Flash) .	121
Instalando um sistema de arquivos raiz ZFS (Instalação com JumpStart)	126
Palavras-chave do JumpStart para ZFS	126

Exemplos de perfil do JumpStart para ZFS	128
Problemas do JumpStart ZFS	129
Migrando um Sistema de Arquivos de Raiz do ZFS ou Atualizando um Sistema de Arquivos de Raiz do ZFS (Live Upgrade)	129
Problemas de migração ZFS com o Live Upgrade	131
Usando o Live Upgrade para migrar ou atualizar um sistema de arquivos raiz do ZFS (sem zonas)	132
Usando o Live Upgrade para migrar ou atualizar um sistema com zonas (Solaris 10 10/08)	139
Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas (no mínimo Solaris 10 5/09)	144
Gerenciamento dos dispositivos de permuta e despejo do ZFS	154
Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS	156
Personalizando Volumes de Permuta e Dump do ZFS	157
Solução de problemas para dispositivo de despejo ZFS	158
Inicializando a partir de um sistema de arquivos raiz ZFS	159
Inicializando a partir de um disco alternativo em um pool raiz ZFS espelhado	159
SPARC: inicializando a partir de um sistema de arquivos raiz ZFS	160
x86: inicializando a partir de um sistema de arquivos raiz ZFS	162
Resolvendo problemas do ponto de montagem ZFS que impedem a inicialização com êxito (Solaris 10 10/08)	163
Inicialização para fins de recuperação em um ambiente raiz ZFS	164
Recuperando o pool raiz ZFS ou instantâneos do pool raiz	166
▼ Como substituir um disco no pool raiz ZFS	166
▼ Como criar instantâneos do pool raiz	169
▼ How to Re-create a ZFS Root Pool and Restore Root Pool Snapshots	170
▼ Como reverter os instantâneos do pool raiz a partir da inicialização à prova de falhas	172
5 Gerenciando sistemas de arquivos ZFS do Oracle Solaris	175
Gerenciando de sistemas de arquivos ZFS (visão geral)	175
Criando, destruindo e renomeando sistemas de arquivos ZFS	176
Criando um sistema de arquivos ZFS	176
Destruindo um sistema de arquivos ZFS	177
Renomeando um sistema de arquivos ZFS	178
Introduzindo as propriedades do ZFS	179
Propriedades nativas somente leitura do ZFS	187

Propriedades nativas definíveis do ZFS	188
Propriedades de usuário do ZFS	191
Consultando informações sobre o sistema de arquivos ZFS	192
Listando informações básicas do ZFS	192
Criando consultas complexas de ZFS	193
Gerenciando propriedades do ZFS	194
Definindo propriedades do ZFS	195
Herdando propriedades do ZFS	195
Consultando propriedades do ZFS	196
Montando sistemas de arquivos ZFS	200
Gerenciando pontos de montagem do ZFS	200
Montando sistemas de arquivos ZFS	202
Usando propriedades de montagem temporárias	203
Desmontando sistemas de arquivos	204
Compartilhando e descompartilhando sistemas de arquivos ZFS	204
Definindo cotas e reservas do ZFS	206
Definindo cotas em sistemas de arquivos ZFS	207
Definindo reservas nos sistemas de arquivos ZFS	210
Atualização de sistemas de arquivos ZFS	212
 6 Trabalhando com instantâneos e clones do Oracle Solaris ZFS	213
Visão geral dos instantâneos do ZFS	213
Criando e destruindo instantâneos do ZFS	214
Exibindo e acessando instantâneos do ZFS	217
Retornando um instantâneo ZFS	218
Identificação de diferenças do instantâneo do ZFS (<code>zfs diff</code>)	219
Visão geral dos clones do ZFS	220
Criando um clone do ZFS	221
Destruindo um clone do ZFS	221
Substituindo um sistema de arquivos ZFS por um clone do ZFS	222
Enviando e recebendo dados do ZFS	223
Salvando dados do ZFS com outros produtos de backup	224
Identificando fluxos de instantâneos do ZFS	224
Enviando um instantâneo do ZFS	226
Recebendo um instantâneo do ZFS	227

Aplicação de valores de propriedade diferentes a um fluxo de instantâneos do ZFS	228
Enviando e recebendo fluxos complexos de instantâneos do ZFS	231
Replicação remota de dados do ZFS	233
7 Uso de ACLs e atributos para proteger arquivos do Oracle Solaris ZFS	235
Modelo de ACL do Solaris	235
Descrições de sintaxe para definição de ACLs	237
Herança da ACL	240
Propriedades da ACL	241
Definindo ACLs em arquivos ZFS	242
Definindo e exibindo ACLs em arquivos ZFS no formato verboso	245
Definindo a herança da ACL em arquivos ZFS no formato verboso	249
Definindo e exibindo ACLs em arquivos ZFS no formato compacto	255
8 Administração delegada do ZFS do Oracle Solaris	261
Visão geral da administração delegada de ZFS	261
Desativando permissões delegadas do ZFS	262
Delegando permissões do ZFS	262
Delegação de permissões do ZFS (zfs allow)	265
Removendo as permissões delegadas do ZFS (zfs unallow)	266
Delegando permissões do ZFS (Exemplos)	266
Exibindo permissões delegadas do ZFS (Exemplos)	270
Removendo permissões delegadas ZFS (exemplos)	272
9 Tópicos avançados do Oracle Solaris ZFS	275
Volumes ZFS	275
Usando um volume ZFS como um dispositivo de permuta ou despejo	276
Usando um volume ZFS como um destino iSCSI Solaris	277
Usando ZFS em um sistema Solaris com zonas instaladas	278
Adicionando sistemas de arquivos ZFS a uma zona não global	279
Delegando conjuntos de dados a uma zona não global	280
Adicionando volumes ZFS a uma zona não global	281
Usando pools de armazenamento do ZFS dentro de uma zona	282
Gerenciando propriedades do ZFS dentro de uma zona	282

Entendendo a propriedade zoned	283
Usando pools de raiz alternativa do ZFS	284
Criando pools de raiz alternativa do ZFS	284
Importando pools de raiz alternativa	285
10 Solução de problemas e recuperação de pools do Oracle Solaris ZFS	287
Identificando Problemas no ZFS	287
Resolvendo Problemas Gerais de Hardware	288
Identificando Falhas de Hardware e Dispositivo	288
Relatório de mensagens de erros do ZFS do sistema	289
Identificando Problemas com Pools de Armazenamento do ZFS	290
Determinando se há problemas em um pool de armazenamento do ZFS	291
Revisando a saída de <code>zpool status</code>	292
Resolvendo Problemas do Dispositivo de Armazenamento do ZFS	295
Resolvendo um Dispositivo Ausente ou Removido	295
Substituindo ou reparando um dispositivo modificado	298
Resolvendo Problemas do Sistema de Arquivos do ZFS	308
Resolvendo Problemas de Dados em um Pool de Armazenamento do ZFS.	308
Verificando a integridade do sistema de arquivos ZFS	309
Dados corrompidos do ZFS	311
Resolvendo Problemas de Espaço do ZFS	311
Reparando Dados Danificados	313
Reparando uma configuração do ZFS danificada	318
Reparando um sistema não inicializável	318
11 Práticas Recomendadas do Oracle Solaris ZFS	321
Práticas Recomendadas do Pool de Armazenamento	321
Práticas Gerais do Sistema	321
Práticas de Criação do Pool de Armazenamento do ZFS	323
Práticas do Pool de Armazenamento para Desempenho	326
Práticas de Monitoramento e Manutenção do Pool de Armazenamento do ZFS	327
Práticas Recomendadas do Sistema de Arquivos	328
Práticas para Criação do Sistema de Arquivos	328
‘Práticas para Monitoramento do Sistema de Arquivos do ZFS	329

- A Descrição das versões do Oracle Solaris ZFS 331**
 - Visão geral das versões do ZFS 331
 - Versões do Pool do ZFS 331
 - Versões do sistema de arquivos do ZFS 333

- Índice 335**

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 usam as famílias SPARC e x86 de arquiteturas de processadores. Os sistemas suportados aparecem na *Oracle Solaris Hardware Compatibility List* em <http://www.oracle.com/webfolder/technetwork/hcl/index.html>. Este documento cita todas as diferenças de implementação entre os tipos de plataformas.

Quem deve usar este manual

Este guia destina-se àqueles que se interessam pela configuração e o gerenciamento dos sistemas de arquivos do Oracle Solaris ZFS. Recomenda-se ter experiência com o sistema operacional (SO) 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 do 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 pools e sistemas de arquivos simples. Este capítulo oferece também o hardware e o software necessários para criar sistemas de arquivos ZFS.
Capítulo 3, “Gerenciando pools de armazenamento do Oracle Solaris ZFS”	Oferece uma descrição detalhada de como criar e administrar pools de armazenamento do ZFS.

Capítulo	Descrição
Capítulo 4, “Instalando e inicializando um sistema de arquivos raiz do Oracle Solaris ZFS”	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 5, “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 6, “Trabalhando com instantâneos e clones do Oracle Solaris ZFS”	Descreve como criar e administrar clones e instantâneos do ZFS.
Capítulo 7, “Uso de ACLs e atributos para proteger arquivos do Oracle Solaris ZFS”	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 8, “Administração delegada do ZFS do Oracle Solaris”	Descreve como utilizar a administração delegada do ZFS para permitir que usuários não-privilegiados efetuem tarefas de administração do ZFS.
Capítulo 9, “Tópicos avançados do Oracle Solaris ZFS”	Fornecer informações sobre o uso dos volumes do ZFS, sobre o uso do ZFS em um sistema Oracle Solaris com zonas instaladas e pools de raiz alternativa.
Capítulo 10, “Solução de problemas e recuperação de pools do Oracle Solaris ZFS”	Descreve como identificar falhas do ZFS e como realizar recuperações a partir delas. Abrange também medidas para evitar falhas.
Capítulo 11, “Práticas Recomendadas do Oracle Solaris ZFS”	Descreve práticas recomendadas para a criação, o monitoramento e a manutenção dos pools de armazenamento e sistemas de arquivos do ZFS.
Apêndice A, “Descrição das versões do Oracle Solaris ZFS”	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: Advanced Administration*
- *System Administration Guide: Devices and File Systems*
- *System Administration Guide: Security Services*

Acesso ao suporte Oracle

Os clientes Oracle possuem acesso a suporte eletrônico por meio do My Oracle Support. Para obter informações, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> se você é portador de deficiência auditiva.

Convenções tipográficas

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

TABELA P-1 Convenções tipográficas

Fonte	Descrição	Exemplo
AaBbCc123	Nomes de comandos, arquivos, diretórios e saídas do computador na tela	Edite seu arquivo <code>.login</code> . Use <code>ls -a</code> para listar todos os arquivos. <code>machine_name%</code> , você tem e-mail.
AaBbCc123	O que você digita, em comparação com a saída do computador na tela	<code>machine_name% su</code> Senha:
<i>aabbcc123</i>	Espaço reservado: substitua, aplicando um nome ou valor real	O comando para remover um arquivo é <code>rm filename</code> .
<i>AaBbCc123</i>	Títulos de manuais, termos novos e termos a serem enfatizados	Consulte 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 os prompts do sistema UNIX e os prompts de superusuário para shells incluídos no SO Oracle Solaris. Nos exemplos de comando, o prompt de shell indica se o comando deve ser executado por um usuário comum ou por um usuário com privilégios.

TABELA P-2 Prompts de shell

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

Sistema de arquivos do Oracle Solaris ZFS (introdução)

Este capítulo oferece uma visão geral do sistema de arquivos do 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 15
- “O que é Oracle Solaris ZFS?” na página 25
- “Terminologia do ZFS” na página 28
- “Requisitos para nomeação de componentes do ZFS” na página 30
- “Diferenças entre o sistema de arquivos tradicional e o Oracle Solaris ZFS” na página 31

O que há de novo no ZFS?

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

- “Aprimoramentos de Uso do Comando ZFS” na página 16
- “Aprimoramento de Instantâneo do ZFS” na página 16
- “Propriedade `aclmode` Aprimorada” na página 17
- “Recursos de instalação do Oracle Solaris ZFS” na página 17
- “Aprimoramentos no fluxo de envio do ZFS” na página 18
- Diferenças do instantâneo do ZFS (“Diferenças do instantâneo do ZFS (`zfs diff`)” na página 18)
- “Recuperação do pool de armazenamento do ZFS e aprimoramentos no desempenho” na página 18
- “Ajuste do comportamento síncrono do ZFS” na página 19
- “Mensagens aprimoradas do pool ZFS” na página 19
- “Aprimoramentos de interoperabilidade ACL do ZFS” na página 20
- “Dividindo um pool de armazenamento do ZFS espelhado (`zpool split`)” na página 22
- “Novo processo do sistema ZFS” na página 22
- “Suporte de instalação do ZFS e Flash” na página 24

- “Gerenciamento do ZFS baseado na Web” na página 24

Aprimoramentos de Uso do Comando ZFS

Oracle Solaris 10 1/13: Os comandos `zfs` e `zpool` têm um subcomando `help` usado para oferecer mais informações sobre os subcomandos `zfs` e `zpool` e suas opções com suporte. Por exemplo:

```
# zfs help
The following commands are supported:
allow      clone      create      destroy     diff        get
groupspace help      hold        holds       inherit     list
mount      promote    receive     release     rename      rollback
send       set         share       snapshot    unallow     unmount
unshare    upgrade    userspace
For more info, run: zfs help <command>
# zfs help create
usage:
    create [-p] [-o property=value] ... <filesystem>
    create [-ps] [-b blocksize] [-o property=value] ... -V <size> <volume>

# zpool help
The following commands are supported:
add      attach  clear  create  destroy  detach  export  get
help     history import  iostat  list     offline online  remove
replace  scrub   set    split   status   upgrade
For more info, run: zpool help <command>
# zpool help attach
usage:
    attach [-f] <pool> <device> <new-device>
```

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

Aprimoramento de Instantâneo do ZFS

Oracle Solaris 10 1/13: esta versão inclui os seguintes aprimoramentos de instantâneo do ZFS:

- O comando `zfs snapshot` tem um alias `snap` que oferece a sintaxe abreviada desse comando. Por exemplo:

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

- O comando `zfs diff` oferece uma opção de enumeração, `-e`, para identificar todos os arquivos adicionados ou modificados entre os dois instantâneos. A saída gerada identifica todos os arquivos adicionados, mas não oferece exclusões possíveis. Por exemplo:

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

Você também pode usar a opção `-o` para identificar os campos selecionados para exibição. Por exemplo:


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

Para obter informações sobre a criação de instantâneos do ZFS, consulte o [Capítulo 6](#), “Trabalhando com instantâneos e clones do Oracle Solaris ZFS”.

Propriedade `aclmode` Aprimorada

Oracle Solaris 10 1/13: A propriedade `aclmode` modifica o comportamento da ACL (Lista de Controle de Acesso) sempre que as permissões de ACL de um arquivo forem modificadas durante uma operação de `chmod`. A propriedade `aclmode` foi introduzida novamente com os seguintes valores:

- `discard` – Um sistema de arquivos com uma propriedade `aclmode` de `discard` excluir todas as entradas de ACL que não representam o modo do arquivo. Este é o valor padrão.
- `mask` – Um sistema de arquivos com uma propriedade `aclmode` de `mask` reduz as permissões de usuário ou grupo. As permissões são reduzidas, de forma que não são maiores que os bits de permissão do grupo, a menos que seja uma entrada de usuário com o mesmo UID do proprietário do arquivo ou diretório. Nesse caso, as permissões de ACL são reduzidas para que não fiquem maiores que os bits da permissão do proprietário. O valor da máscara também preserva a ACL nas alterações de modo, desde que uma operação explícita de definição da ACL não tenha sido executada.
- `passthrough` – Um sistema de arquivos com uma propriedade `aclmode` de `passthrough` indica que não houve alterações na ACL além da geração das entradas de ACL necessárias para representar o novo modo do arquivo ou diretório.

Para obter mais informações, consulte o [Exemplo 7–13](#).

Recursos de instalação do Oracle Solaris ZFS

Oracle Solaris 10 8/11: Nesta versão, os novos recursos de instalação a seguir estão disponíveis:

- Você pode usar o método de instalação no modo de texto para instalar um sistema com um arquivo flash ZFS. Para obter mais informações, consulte o [Exemplo 4–3](#).
- Você pode usar o comando `luupgrade` do Oracle Solaris Live Upgrade para instalar um arquivo flash raiz ZFS. Para obter mais informações, consulte o [Exemplo 4–8](#).
- Você pode usar o comando `lucreate` do Oracle Solaris Live Upgrade para especificar um sistema de arquivos `/var` separado. Para obter mais informações, consulte o [Exemplo 4–5](#).

Aprimoramentos no fluxo de envio do ZFS

Oracle Solaris 10 8/11: Nesta versão, você pode definir propriedades do sistema de arquivos que são enviadas e recebidas em um fluxo de instantâneo. Esses aprimoramentos oferecem flexibilidade na aplicação de propriedades do sistema em um fluxo de envio para o sistema de arquivos de recebimento ou ao determinar se as propriedades do sistema de arquivos local, como o valor de propriedade `mountpoint`, devem ser ignorados quando recebidos.

Para obter mais informações, consulte [“Aplicação de valores de propriedade diferentes a um fluxo de instantâneos do ZFS”](#) na página 228.

Diferenças do instantâneo do ZFS (`zfs diff`)

Oracle Solaris 10 8/11: Nesta versão, você pode determinar diferenças do instantâneo ZFS usando o comando `zfs diff`.

Por exemplo, considere que estes dois instantâneos sejam criados:

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

Por exemplo, para identificar as diferenças entre dois instantâneos, use sintaxe semelhante à seguinte:

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

Na saída, `M` indica que o diretório foi modificado. `+` indica que `fileB` existe no instantâneo posterior.

Para obter mais informações, consulte [“Identificação de diferenças do instantâneo do ZFS \(`zfs diff`\)”](#) na página 219.

Recuperação do pool de armazenamento do ZFS e aprimoramentos no desempenho

Oracle Solaris 10 8/11: Nesta versão, os seguintes novos recursos do pool de armazenamento ZFS são fornecidos:

- Você pode importar um pool com um log ausente usando o comando `zpool import -m`. Para obter mais informações, consulte [“Importação de um pool com um dispositivo de log ausente”](#) na página 102.

- Você pode importar um pool no modo somente leitura. Este recurso é principalmente para recuperação do pool. Se um pool danificado não puder ser acessado porque os dispositivos subjacentes estão danificados, você poderá importar o pool somente leitura para recuperar os dados. Para obter mais informações, consulte [“Importação de um pool no modo somente leitura” na página 104](#).
- Um pool de armazenamento RAID-Z (raidz1, raidz2 ou raidz3) que é criado nesta versão terá alguns metadados sensíveis à latência espelhados automaticamente para aprimorar o desempenho de transferência de E/S de leitura. Para pools RAID-Z existentes atualizados para pelo menos o pool versão 29, alguns metadados serão espelhados para todos os dados recém-gravados.

Os metadados espelhados em um pool RAID-Z *não* fornece proteção adicional contra falhas de hardware, semelhante ao que um pool de armazenamento espelhado oferece. Espaço adicional é consumido por metadados espelhados, mas a proteção RAID-Z permanece igual à de versões anteriores. Este aprimoramento é para fins de desempenho apenas.

Ajuste do comportamento síncrono do ZFS

Solaris 10 8/11: Nesta versão, você pode determinar o comportamento síncrono de um sistema de arquivos ZFS usando a propriedade sync.

O comportamento síncrono padrão é gravar todas as transações do sistema de arquivos síncronas no log de intenção e descarregar todos os dispositivos para garantir que os dados estejam estáveis. A desativação do comportamento síncrono padrão não é recomendada. Aplicativos que dependem de suporte síncrono podem ser afetados e pode ocorrer perda de dados.

A propriedade sync pode ser configurada antes ou após a criação do sistema de arquivos. Em qualquer dos casos, o valor de propriedade é efetivado imediatamente. Por exemplo:

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

O parâmetro `zil_disable` não está mais disponível nas versões do Oracle Solaris que incluem a propriedade sync.

Para obter mais informações, consulte a [Tabela 5–1](#).

Mensagens aprimoradas do pool ZFS

Oracle Solaris 10 8/11: Nesta versão, você pode usar a opção -T para oferecer um intervalo e valor de contagem para os comandos `zpool list` e `zpool status` para exibir informações adicionais.

Além disso, mais informações de scrub (escovação) e resilver (polimento) do pool são fornecidas pelo comando `zpool status`, desta forma:

- Relatório de andamento de resilver. Por exemplo:


```
scan: resilver in progress since Thu Jun  7 14:41:11 2012
      3.83G scanned out of 73.3G at 106M/s, 0h11m to go
      3.80G resilvered, 5.22% done
```
- Relatório de andamento de scrub. Por exemplo:


```
scan: scrub in progress since Thu Jun  7 14:59:25 2012
      1.95G scanned out of 73.3G at 118M/s, 0h10m to go
      0 repaired, 2.66% done
```
- Mensagem de conclusão de resilver. Por exemplo:


```
resilvered 73.3G in 0h13m with 0 errors on Thu Jun  7 14:54:16 2012
```
- Mensagem de conclusão de scrub. Por exemplo:


```
scan: scrub repaired 512B in 1h2m with 0 errors on Thu Jun  7 15:10:32 2012
```
- Mensagem de cancelamento de scrub em andamento. Por exemplo:


```
scan: scrub canceled on Thu Jun  7 15:19:20 MDT 2012
```
- Mensagens de conclusão de scrub e de resilver persistem nas reinicializações do sistema

A sintaxe a seguir usa o intervalo e a opção de contagem para exibir informações constantes de resilver do pool. Você pode usar o valor `-T d` para exibir as informações no formato de data padrão ou `-T u` para exibir as informações em um formato interno.

```
# zpool status -T d tank 3 2
Wed Nov 14 15:44:34 MST 2012
pool: tank
state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
       continue to function in a degraded state.
action: Wait for the resilver to complete.
scan: resilver in progress since Wed Nov 14 15:44:34 2012
      2.96G scanned out of 4.19G at 189M/s, 0h0m to go
      1.48G resilvered, 70.60% done
config:

    NAME                                STATE      READ WRITE CKSUM
    tank                                DEGRADED   0     0     0
      mirror-0
        c0t5000C500335F95E3d0          ONLINE    0     0     0
        c0t5000C500335F907Fd0          ONLINE    0     0     0
      mirror-1
        c0t5000C500335BD117d0          ONLINE    0     0     0
        c0t5000C500335DC60Fd0          DEGRADED   0     0     0 (resilvering)

errors: No known data errors
```

Aprimoramentos de interoperabilidade ACL do ZFS

Oracle Solaris 10 8/11: Nesta versão, os seguintes aprimoramentos da ACL são fornecidos:

- ACLs triviais não exigem ACEs (entradas de controle de acesso) deny, exceto para permissões não usuais. Por exemplo, um modo de 0644, 0755 ou 0664 não requer ACEs deny, mas um modo como 0705, 0060 e assim por diante requer ACEs deny.

O comportamento antigo inclui ACEs deny m qualquer ACL trivial, como 644. Por exemplo:

```
# ls -v file.1
-rw-r--r-- 1 root    root      206663 Jun 14 11:52 file.1
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

O novo comportamento para uma ACL trivial, como 644, não inclui as ACEs deny. Por exemplo:

```
# ls -v file.1
-rw-r--r-- 1 root    root      206663 Jun 22 14:30 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

- ACLs não são mais divididas em várias ACEs durante a herança para tentar preservar a permissão não modificada original. Em vez disso, as permissões são modificadas conforme necessário para impor o modo de criação de arquivo.
- O comportamento da propriedade `aclinherit` inclui uma redução de permissões quando a propriedade é definida como `restricted`, o que significa que as ACLs não são mais divididas em várias ACEs durante a herança.
- Uma nova regra de cálculo do modo de permissão específica que, se uma ACL tiver uma ACE de *usuário* que também seja o proprietário do arquivo, então essas permissões serão incluídas no cálculo do modo de permissão. A mesma regra será aplicada se uma ACE ao *grupo* for o proprietário do grupo do arquivo.

Para obter mais informações, consulte [Capítulo 7, “Uso de ACLs e atributos para proteger arquivos do Oracle Solaris ZFS”](#).

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

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

Para obter mais informações, consulte [“Criando um novo pool ao dividir um pool de armazenamento espelhado do ZFS” na página 69.](#)

Novo processo do sistema ZFS

Oracle Solaris 10 9/10 : nessa versão, cada pool de armazenamento do ZFS possui um processo associado, `zpool -poolname`. Os threads nesse processo são os threads de processamento de E/S do pool que manipulam tarefas de E/S, tais como compactação e validação da soma de verificação, que estão associadas ao pool. O propósito deste processo é fornecer visibilidade em cada utilização da CPU do pool de armazenamento.

Informações sobre estes processos em execução podem ser revisadas utilizando os comandos `ps` e `prstat`. Estes processos só estão disponíveis em uma zona global. Para obter mais informações, consulte [SDC\(7\)](#).

Aprimoramentos de substituição de dispositivo do ZFS

Oracle Solaris 10 9/10: Nesta versão, um evento do sistema ou *sysevent* é fornecido quando os discos em um pool são substituídos por discos maiores. O ZFS foi aprimorado para reconhecer esses eventos e ajustar o pool com base no novo tamanho do disco, dependendo da configuração da propriedade `autoexpand`. Você pode usar a propriedade do pool `autoexpand` para ativar ou desativar a expansão automática do pool quando um disco maior substituir um disco menor.

Esses aprimoramentos permitem aumentar o tamanho do pool sem precisar exportar e importar o pool ou reinicializar o sistema.

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

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

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

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

A propriedade `autoexpand` é desativada por padrão, portanto você pode decidir se deseja que o tamanho do pool seja expandido quando um disco maior substituir um disco menor.

O tamanho do pool também pode ser expandido utilizando o comando `zpool online -e`. Por exemplo:

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

É possível redefinir a propriedade `autoexpand` depois de o LUN ser anexado ou disponibilizado com a utilização do comando `zpool replace`. Por exemplo, o pool 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 pool não é expandido até que a propriedade `autoexpand` seja ativada.

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

Outra maneira de expandir o disco sem ativar a propriedade `autoexpand` é utilizar o comando `zpool online -e` mesmo que o dispositivo já esteja on-line. Por exemplo:

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

Os aprimoramentos adicionais de substituição de dispositivo nesta versão incluem o seguinte:

- Em versões anteriores, o ZFS não conseguia 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 quase do mesmo tamanho, desde que o pool não esteja cheio.
- Nesta versão, não é necessário reinicializar o sistema ou exportar e importar um pool para expandir o tamanho do pool. Conforme descrito anteriormente, você pode ativar a propriedade `autoexpand` ou utilizar o comando `zpool online -e` para expandir o tamanho do pool.

Para obter informações sobre a substituição de dispositivos, consulte [“Substituindo dispositivos em um pool de armazenamento” na página 75](#).

Suporte de instalação do ZFS e Flash

Solaris 10 10/09: nessa versão, é possível configurar um perfil JumpStart para identificar um arquivo Flash de um pool raiz ZFS. Para mais informações, consulte [“Instalando um sistema de arquivos raiz ZFS \(instalação de arquivo do Oracle Solaris Flash\)”](#) na página 121.

Migração de zonas em um ambiente do ZFS

Solaris 10 5/09: essa versão amplia o suporte à migração de zonas em um ambiente ZFS com o Oracle Solaris Live Upgrade. Para mais informações, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas \(no mínimo Solaris 10 5/09\)”](#) na página 144.

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

Suporte à inicialização e instalação do ZFS

Solaris 10 10/08: essa versão permite que você instale e inicialize um sistema de arquivos raiz 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 UFS para um sistema de arquivos raiz ZFS. O suporte do ZFS para dispositivos de permuta e despejo também é oferecido. Para mais informações, consulte [Capítulo 4, “Instalando e inicializando um sistema de arquivos raiz do Oracle Solaris ZFS”](#).

Para obter uma lista dos problemas conhecidos desta versão, consulte as notas de versão do Solaris 10 10/08.

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 Oracle Solaris 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 forçar você a criar volumes virtualizados, o ZFS agrega os dispositivos em um pool de armazenamento. O pool 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 pool. 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 pool de armazenamento. Quando novos armazenamentos são adicionados, todos os sistemas de arquivos que estão dentro do pool podem utilizar imediatamente o espaço adicional sem trabalho adicional. Em muitos aspectos, o pool de armazenamento trabalha de maneira semelhante 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 sequência 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 detectados e possam ser recuperados adequadamente. A verificação da soma de verificação e a recuperação de dados são executadas na camada do sistema de arquivos e são transparentes para os aplicativos.

Além disso, o ZFS oferece a autocorreção de dados. O ZFS suporta pools 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 sistema de arquivos ou volume. Os instantâneos podem ser criados rápida e facilmente. No início, os instantâneos não utilizam espaço adicional em disco dentro do pool.

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 220 .
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 único nome no espaço de nome do ZFS. Os conjuntos de dados são identificados com o seguinte formato: <i>pool/path[@snapshot]</i> <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
sistema de arquivos		<p>Para mais informações sobre conjuntos de dados, consulte o Capítulo 5, “Gerenciando sistemas de arquivos ZFS do Oracle Solaris”.</p> <p>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.</p> <p>Para mais informações sobre sistemas de arquivos, consulte o Capítulo 5, “Gerenciando sistemas de arquivos ZFS do Oracle Solaris”.</p>
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 pool.
ambiente de inicialização principal		<p>Para obter mais informações sobre pools de armazenamento, consulte o Capítulo 3, “Gerenciando pools de armazenamento do Oracle Solaris ZFS”.</p> <p>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>.</p>
RAID-Z		Um dispositivo virtual que armazena dados e paridades em discos múltiplos. Para obter mais informações sobre o RAID-Z, consulte “Configuração de pool de armazenamento RAID-Z” na página 47 .
resilvering		O processo de cópia de dados de um dispositivo para outro é conhecido como <i>resilvering</i> . Por exemplo, se

	<p>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 obter mais informações sobre resilvering do ZFS, consulte “Exibindo o status do resilvering” na página 307.</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 obter informações sobre instantâneos, consulte “Visão geral dos instantâneos do ZFS” na página 213.</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 obter mais informações sobre dispositivos virtuais, consulte “Exibindo informações do dispositivo virtual do pool de armazenamento” na página 57.</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 intercâmbio.</p> <p>Para obter informações sobre volumes ZFS, consulte “Volumes ZFS” na página 275.</p>

Requisitos para nomeação de componentes do ZFS

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

- Cada componente pode conter somente caracteres alfanuméricos além destes quatro caracteres especiais:
 - Sublinhado (_)
 - Hífen (-)
 - Dois-pontos (:)
 - Ponto final (.)

- Os nomes de pool devem começar com uma letra e só podem conter caracteres alfanuméricos, além de sublinhado (`_`), travessão (`-`) e ponto final (`.`). Observe as seguintes restrições de nome de pool:
 - A sequência inicial `c[0-9]` não é permitida.
 - O nome `log` é reservado.
 - Um nome que começa com `mirror`, `raidz`, `raidz1`, `raidz2`, `raidz3` ou `spare` não é permitido porque esses nomes são reservados.
 - Os nomes de pool 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.

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

- [“Granularidade do sistema de arquivos ZFS” na página 31](#)
- [“Contabilidade de espaço em disco do ZFS” na página 32](#)
- [“Montando sistemas de arquivos ZFS” na página 34](#)
- [“Gerenciamento de volume tradicional” na página 34](#)
- [“Modelo de ACL do Solaris Baseado em NFSv4” na página 34](#)

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 pool 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 39.](#)

Contabilidade de espaço em disco do ZFS

O ZFS é baseado no conceito de armazenamento em pools. 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 pool 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 207.](#) 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 210.](#) 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.

Em um sistema de arquivos UFS, o comando `du` reporta o tamanho do bloco de dados no arquivo. Em um sistema ZFS, `du` reporta o tamanho real do arquivo como armazenado no disco. Esse tamanho inclui metadados e compactação. Esse relatório ajuda a responder "quanto mais de espaço obterei se remover este arquivo?" Portanto, mesmo quando a compactação está desativada, você ainda verá resultados diferentes entre ZFS e UFS.

Quando você compara o consumo do espaço que é reportado pelo comando `df` com o comando `zfs list`, considere que `df` está reportando o tamanho do pool e não apenas os tamanhos dos sistemas de arquivos. Além disso, `df` não entende sistemas de arquivo descendentes ou se existem instantâneos. Se alguma propriedade de ZFS, como compactação e cotas, for definida nos sistemas de arquivos, a reconciliação de consumo de espaço reportada pelo `df` poderá ser difícil.

Considere os seguintes cenários que também podem afetar o consumo de espaço informado:

- Para os arquivos maiores que `recordsize`, o último bloco de arquivos geralmente fica metade cheio. Com o `recordsize` padrão definido como 128 KB, aproximadamente 64 KB são perdidos por arquivo, o que pode ser um grande impacto. A integração de RFE 6812608 resolve este cenário. Você pode solucionar isso ativando a compactação. Mesmo se seus dados já estiverem compactados, a parte não usada do último bloco será preenchida com zeros e a compactação será muito boa.
- Em um pool RAIDZ-2, cada bloco consome pelo menos 2 setores (porções de 512 bytes) de informações de paridade. O espaço consumido pelas informações de paridade não é reportado, mas como pode variar e ser uma porcentagem muito maior para blocos pequenos, talvez seja observado um impacto no relatório de espaços. O impacto é mais extremo para um `recordsize` definido como 512 bytes, em que cada bloco lógico de 512 bytes consome 1,5 KB (3 vezes o espaço). Independentemente dos dados que estão sendo armazenados, se a eficiência de espaço for sua principal preocupação, você deve manter o `recordsize` como padrão (128 KB) e ativar a compactação (para o padrão de `lzjb`).
- O comando `df` não conhece os dados de arquivos deduplicados.

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 6, “Trabalhando com instantâneos e clones do Oracle Solaris ZFS”](#).

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 sistema de arquivos. 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 sistemas de arquivos ZFS” na página 200](#).

Gerenciamento de volume tradicional

De acordo com o descrito em [“Armazenamento de ZFS em pool” na página 25](#), 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.

Modelo de ACL do Solaris Baseado em NFSv4

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 7, “Uso de ACLs e atributos para proteger arquivos do Oracle Solaris ZFS”](#).

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:

- “Perfis de direitos do ZFS” na página 35
- “Recomendações e requisitos de software e hardware do ZFS” na página 36
- “Criando um sistema de arquivos ZFS básico” na página 36
- “Criando um pool de armazenamento básico do ZFS” na página 37
- “Criando uma hierarquia de sistemas de arquivos ZFS” na página 39

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 pool 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 [Capítulo 8, “Administração delegada do ZFS do Oracle Solaris”](#).

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:

- Use um sistema com base em SPARC ou x86 que esteja executando uma versão do Oracle Solaris com suporte.
- O espaço mínimo em disco necessário para um pool de armazenamento é de 64 MB. O tamanho mínimo do disco é de 128 MB.
- Para obter um bom desempenho de ZFS, dimensione os requisitos de memória com base na sua carga de trabalho.
- Se você criar uma configuração de pool 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 pool, um novo sistema de arquivos do ZFS é automaticamente criado e montado.

O exemplo a seguir mostra como criar um pool 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 47](#).

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 pool. Os sistemas de arquivos fornecem pontos de administração que permitem gerenciar diferentes conjuntos de dados dentro de um mesmo pool.

O exemplo a seguir mostra como criar um sistema de arquivos chamado `fs` no pool 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 39](#).

Criando um pool de armazenamento básico 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 pool 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 38](#), 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 43](#).

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 pool 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 38](#), 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 47](#).

▼ 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 35](#).

2 Escolha um nome para o pool de armazenamento.

Esse nome é utilizado para identificar o pool de armazenamento quando os comandos `zpool` ou `zfs` estão sendo utilizados. Selecione qualquer nome de pool de sua preferência, mas ele deve satisfazer os requisitos de nomeação dos [“Requisitos para nomeação de componentes do ZFS” na página 30](#).

3 Crie o pool.

Por exemplo, o comando a seguir cria um pool 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 Pools de Armazenamento do ZFS” na página 50](#). Para obter mais informações sobre como é determinado o uso do dispositivo, consulte [“Detectando dispositivos em uso” na página 58](#).

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 86](#).

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, `jeff` e `bill`, são criados em [“Como criar sistemas de arquivos ZFS” na página 40](#).

Para obter mais informações sobre o gerenciamento de sistemas de arquivos, consulte [Capítulo 5, “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 40](#), 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 40](#), todos os diretórios `home` estão montados no usuário `/export/zfs/`, compartilhados através do NFS e apresentam a compactação ativada. Além disso, no usuário `jeff`, é obrigatória uma cota de 10 GB.

Para obter mais informações sobre propriedades, consulte [“Introduzindo as propriedades do ZFS” na página 179](#).

▼ 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 35](#).

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 ele 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 179](#).

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

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 176](#).

Para obter mais informações sobre a montagem e o compartilhamento de sistemas de arquivos ZFS, consulte [“Montando sistemas de arquivos ZFS” na página 200](#).

5 Defina as propriedades específicas do sistema de arquivos.

Neste exemplo, uma cota de 10 GB é atribuída ao usuário `jeff`. 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/jeff
```

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/bill                      8K    67.0G   8K   /export/zfs/bill
tank/home/jeff                      8K    10.0G   8K   /export/zfs/jeff
```

Observe que o usuário `jeff` tem somente 10 GB de espaço disponível, enquanto que o usuário `bill` pode utilizar o pool 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 192](#).

Para obter mais informações sobre como o espaço é utilizado e calculado, consulte [“Contabilidade de espaço em disco do ZFS” na página 32](#).

Gerenciando pools de armazenamento do Oracle Solaris ZFS

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

Este capítulo traz as seguintes seções:

- “Componentes de um pool de armazenamento do ZFS” na página 43
- “Recursos de replicação de um pool de armazenamento do ZFS” na página 47
- “Criando e destruindo pools de armazenamento do ZFS” na página 50
- “Gerenciando dispositivos em pools de armazenamento do ZFS” na página 62
- “Gerenciando propriedades do pool de armazenamento do ZFS” na página 83
- “Consultando status de pool de armazenamento do ZFS” na página 86
- “Migrando pools de armazenamento do ZFS” na página 97
- “Atualizando pools de armazenamento do ZFS” na página 106

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 43
- “Usando faixas em um pool de armazenamento do ZFS” na página 44
- “Usando arquivos em um pool de armazenamento do ZFS” na página 46

Usando discos em um pool de armazenamento do ZFS

O elemento mais básico de um pool 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: 143358287 + 16384 (reserved sectors)

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

Examine as considerações a seguir ao usar discos inteiros nos pools de armazenamento do ZFS:

- Para utilizar um disco inteiro, o disco deve ser nomeado utilizando a convenção de identificação `/dev/dsk/cNtNdN`. 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.
- Em um sistema baseado em x86, o disco deve conter uma partição `fdisk` válida do Solaris. Para obter mais informações sobre a criação ou alteração de uma partição `fdisk` do Solaris, consulte [“Setting Up Disks for ZFS File Systems \(Task Map\)” no System Administration Guide: Devices and File Systems](#).
- 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 \(GPT\) Disk Label” no System Administration Guide: Devices and File Systems](#).

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`

Usando faixas em um pool de armazenamento do ZFS

É possível rotular discos com um rótulo do Solaris VTOC (SMI) quando você cria um pool de armazenamento com um segmento de disco, mas o uso de segmento de discos para um pool não é recomendado porque o gerenciamento desses segmentos de disco é mais difícil.

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 clt1d0
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

Uma partição fdisk também existe em sistemas baseados em x86. Uma partição fdisk é representada por um nome de dispositivo /dev/dsk/cN[tN]dNpN e atua como um contêiner para as partes disponíveis do disco. Não use um dispositivo cN[tN]dNpN para um componente do pool de armazenamento ZFS, porque essa configuração não foi testada nem é suportada.

Usando arquivos em um pool de armazenamento do ZFS

O ZFS também permite que usar arquivos como dispositivos virtuais no pool de armazenamento. Este recurso se destina principalmente a testar e capacitar experimentos simples, não para uso em produção.

- Se criar um pool do ZFS baseado em arquivos em um sistema de arquivos UFS, você estará implicitamente confiando no UFS para garantir correção e semântica síncrona.
- Se você criar um pool do ZFS baseado em arquivos ou volumes criados em outro pool do ZFS, o sistema poderá travar ou entrar em emergência.

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.

Considerações para Pools de Armazenamento do ZFS

Examine as considerações a seguir quando criar e gerenciar pools de armazenamento do ZFS.

- A utilização de discos inteiros é a forma mais simples de criar pools de armazenamento do 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 recomendações do pool de armazenamento, consulte o [Capítulo 11, “Práticas Recomendadas do Oracle Solaris ZFS”](#).
- 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 pool 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 outro alteração de hardware, pode alterar as IDs do dispositivo no pool de armazenamento do ZFS, que pode causar a invalidação do dispositivo.

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 47
- “Configuração de pool de armazenamento RAID-Z” na página 47
- “Dados de autocorreção em uma configuração redundante” na página 49
- “Distribuição dinâmica em um pool de armazenamento” na página 49
- “Pool armazenamento híbrido do ZFS” na página 49

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 50.

Configuração de pool de armazenamento RAID-Z

Além de uma configuração de pool 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.oracle.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 $(N-P) \times X$ 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 e assim por diante. 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 obter informações sobre a criação de um pool de armazenamento RAID-Z, consulte [“Criando um pool de armazenamento RAID-Z” na página 52](#).

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.oracle.com/roch/entry/when_to_and_not_to

Para obter informações adicionais sobre recomendações do pool de armazenamento RAID-Z, consulte o [Capítulo 11, “Práticas Recomendadas do Oracle Solaris ZFS”](#).

Pool armazenamento híbrido do ZFS

O pool de armazenamento híbrido do ZFS, disponível na série de produtos Sun Storage 7000, é um pool de armazenamento especial que combina DRAM, SSDs e HDDs para melhorar o desempenho e aumentar a capacidade, ao mesmo tempo que reduz o consumo de energia. Com esta interface de gerenciamento de produto, é possível selecionar a configuração de redundância do ZFS do pool 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 onde 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 pool, 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 pool. 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 pool, 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 Pools de Armazenamento do ZFS” na página 50
- “Exibindo informações do dispositivo virtual do pool de armazenamento” na página 57
- “Manipulando erros de criação do pool de armazenamento do ZFS” na página 58
- “Destruindo pools de armazenamento do ZFS” na página 61

Criar e destruir pools 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 pool é mais fácil do que construir um. Use `zpool destroy` com cuidado. Este comando simples tem consequências significantes.

Criando Pools 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 pool deve cumprir as convenções de identificação descritas em “Requisitos para nomeação de componentes do ZFS” na página 30.

Criando um pool de armazenamento básico

O seguinte comando cria um novo pool denominado `tank` que contém os discos `clt0d0` e `clt1d0`:

```
# zpool create tank clt0d0 clt1d0
```

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 recomendações de configurações espelhadas, consulte o [Capítulo 11, “Práticas Recomendadas do Oracle Solaris ZFS”](#).

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 62](#).
- 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 67](#).
- 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 75](#).
- 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 67](#).
- Dividindo uma configuração espelhada ao desanexar um dos discos ao criar um novo pool idêntico. Para obter mais informações, consulte [“Criando um novo pool ao dividir um pool de armazenamento espelhado do ZFS” na página 69](#).

Não é possível remover completamente um dispositivo que não é um disco de reserva, um dispositivo de log ou um dispositivo de cache de um pool de armazenamento espelhado.

Criando um pool raiz do ZFS

Considere os seguintes requisitos de configuração do pool raiz:

- O pool 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 pool raiz não pode possuir um dispositivo de log separado.
- Se tentar utilizar uma configuração não suportada para um pool 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 4, “Instalando e inicializando um sistema de arquivos raiz do Oracle Solaris ZFS”](#).

Criando um pool 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 pool. 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 c8t0d0
# zpool status -v tank
pool: tank
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	0	0	0
raidz3-0	ONLINE	0	0	0
c0t0d0	ONLINE	0	0	0
c1t0d0	ONLINE	0	0	0
c2t0d0	ONLINE	0	0	0
c3t0d0	ONLINE	0	0	0

```

c4t0d0 ONLINE      0      0      0
c5t0d0 ONLINE      0      0      0
c6t0d0 ONLINE      0      0      0
c7t0d0 ONLINE      0      0      0
c8t0d0 ONLINE      0      0      0
errors: No known data errors

```

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

- Adicionar outro pool 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 62.](#)
- 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 75.](#)

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

- Anexação de um disco adicional a uma configuração RAID-Z existente.
- Desconectando um disco de uma configuração RAID-Z, exceto quando você está desanexando um disco que foi substituído por um disco de reserva ou quando você precisa desanexar um disco de reserva.
- Não é possível remover completamente um dispositivo que não seja 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 47.](#)

Criando um pool de armazenamento do ZFS com dispositivos de log

O ZIL (ZFS intent log) é fornecido para satisfazer requisitos de POSIX de 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. O NFS e outros aplicativos também podem usar o `fsync()` para garantir a estabilidade de dados.

Por padrão, o ZIL é alocado dos blocos no pool principal. Entretanto, talvez seja possível obter um desempenho melhor com o uso separado de dispositivos de log, como NVRAM ou um disco dedicado.

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

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

- Qualquer melhoria no desempenho visto pela implementação de um dispositivo de log separado depende do tipo de dispositivo, da configuração de hardware do pool da carga de trabalho do aplicativo. Para obter informações preliminares sobre desempenho, consulte este blog:
http://blogs.oracle.com/perrin/entry/slog_blog_or_blogging_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 registro separado não for espelhado e o dispositivo que contém o registro falhar, o armazenamento dos blocos de registro reverterem ao pool de armazenamento.
- Os dispositivos de log podem ser adicionados, substituídos, removidos, anexados, desanexados, importados e exportados como parte do pool de armazenamento maior.
- 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 pool de armazenamento não espelhado.
- O tamanho mínimo de um dispositivo de log é o mesmo do tamanho mínimo de cada dispositivo em um pool, 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.

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

O seguinte exemplo mostra como criar um pool de armazenamento espelhado com dispositivos de log espelhados:

```
# zpool create datap mirror c0t5000C500335F95E3d0 c0t5000C500335F907Fd0 mirror
c0t5000C500335BD117d0 c0t5000C500335DC60Fd0 log mirror c0t5000C500335E106Bd0 c0t5000C500335FC3E7d0
# zpool status datap
pool: datap
state: ONLINE
scrub: none requested
config:
```

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

```
logs
mirror-2          ONLINE      0      0      0
c0t5000C500335E106Bd0  ONLINE      0      0      0
c0t5000C500335FC3E7d0  ONLINE      0      0      0
```

errors: No known data errors

Para obter informações sobre como recuperar de um dispositivo de registro com falha, consulte [Exemplo 10–2](#).

Criando um pool de armazenamento do ZFS com dispositivos de cache

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.

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

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 pool depois dele ser criado.

Considere os seguintes pontos ao determinar a criação de um pool de armazenamento do ZFS com dispositivos de cache:

- A utilização dos dispositivos de cache oferece a maior melhoria de desempenho para cargas de trabalho de leitura aleatória do 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 pool é criado. Também podem ser adicionados e removidos depois da criação do pool. Para obter mais informações, consulte o [Exemplo 3–4](#).
- 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 pool 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.

Cuidados para a Criação de Pools de Armazenamento

Examine os cuidados a seguir quando criar e gerenciar pools de armazenamento do ZFS.

- Não particione ou rotule novamente os discos que são parte de um pool de armazenamento existente. Se você tentar particionar ou rotular novamente um disco de pool de raiz, talvez seja necessário reinstalar o SO.
- Não crie um pool de armazenamento que contenha componentes de outro pool de armazenamento, como arquivos ou volumes. Essa configuração não tem suporte e podem ocorrer travamentos.
- Um pool criado com um único segmento ou único disco não tem redundância e está sob risco de perda de dados. Um pool criado com vários segmentos, mas sem redundância, também está sob risco de perda de dados. Um pool criado com vários segmentos no disco é mais difícil de gerenciar que um pool criado com discos inteiros.
- Um pool que não é criado com redundância de ZFS (RAIDZ ou espelho) só pode reportar inconsistências de dados. Ele não pode reparar inconsistências de dados.
- Embora um pool criado com redundância de ZFS possa ajudar a reduzir o tempo de inatividade devido a falhas de hardware, ele não está imune a falhas de hardware, falhas de alimentação ou cabos desconectados. Faça backup de dados regularmente. É importante realizar backups de rotina em dados do pool de hardwares não corporativos.
- Não é possível compartilhar pools nos sistemas. O ZFS não é um sistema de arquivos em cluster.

Exibindo informações do dispositivo virtual do pool de armazenamento

Cada pool de armazenamento contém um ou mais dispositivos virtuais. Um *dispositivo virtual* é uma representação interna do pool 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 pool 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 disco de reserva é um dispositivo virtual especial que rastreia discos de reserva com reposição em funcionamento disponíveis para um pool.

Os seguintes exemplos mostram como criar um pool 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 pool que consiste em um dispositivo virtual de nível superior de quatro discos:

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

É possível adicionar outro dispositivo virtual de nível superior a este pool 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 pools 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 corrigí-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 umount.
- Sistema de arquivos em /etc/vfstab

O disco contém um sistema de arquivos que está listado no arquivo /etc/vfstab, mas o sistema de arquivos não está montado atualmente. Para corrigir este erro, remova ou anote a linha no arquivo /etc/vfstab.
- 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 dumpadm.

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 `zpool destroy` para destruir o pool, se este não for mais necessário. Ou, use o comando `zpool detach` 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 `-f` 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 `potentially active`, 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 substituí-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 substituir 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 substituir os avisos.

Simulando uma criação de pool de armazenamento

A tentativa de criar um pool 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 pool sem realmente gravar 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 pool quando realmente executado sem esta opção.

Ponto de montagem padrão para pools de armazenamento

Quando um pool é criado, o ponto de montagem padrão para o sistema de arquivos de 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 sistema de arquivos de s 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 pool home e o sistema de arquivos 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 200](#).

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 pool 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 105](#).

Se você destruir um pool com o comando `zpool destroy`, o pool ainda ficará disponível para importação, como descrito em [“Recuperando pools de armazenamento destruídos do ZFS” na página 105](#). Isso significa que os dados confidenciais ainda podem estar disponíveis nos discos que eram parte do pool. Se você quiser destruir dados nos discos destruídos do pool, deverá usar um recurso como a opção `analyze->purge` do utilitário `format` em todos os discos do pool destruído.

Destruindo um pool com dispositivos indisponíveis

O ato de destruir um pool requer que os dados sejam gravados em disco para indicar que o pool 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 pool. Eles aparecem como dispositivos válidos ao buscar por pools para importar. Se um pool apresentar dispositivos UNAVAIL suficientes, de forma que o próprio pool fique UNAVAIL (o que significa que o dispositivo virtual de primeiro nível é UNAVAIL), o comando imprimirá um aviso e não poderá ser concluído sem a opção `-f`. Esta opção é necessária porque o pool 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 92.

Para obter mais informações sobre a importação de pools, consulte [“Importando pools de armazenamento do ZFS”](#) na página 101.

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 43. Depois da criação de um pool, é possível efetuar várias tarefas para gerenciar os dispositivos físicos dentro do pool.

- [“Adicionando dispositivos a um pool de armazenamento”](#) na página 62
- [“Anexando e desanexando dispositivos em um pool de armazenamento”](#) na página 67
- [“Criando um novo pool ao dividir um pool de armazenamento espelhado do ZFS”](#) na página 69
- [“Colocando dispositivos off-line e on-line em um pool de armazenamento”](#) na página 72
- [“Limpendo erros de dispositivo do pool de armazenamento”](#) na página 74
- [“Substituindo dispositivos em um pool de armazenamento”](#) na página 75
- [“Designando discos de reserva com reposição em funcionamento no pool de armazenamento”](#) na página 78

Adicionando dispositivos a um pool de armazenamento

É possível adicionar espaço de disco dinamicamente a um pool 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 pool zeepool.

Para obter mais informações sobre como é realizada a validação de dispositivos virtuais, consulte [“Detectando dispositivos em uso” na página 58](#).

EXEMPLO 3-1 Adicionando discos a uma configuração espelhada do ZFS

No exemplo anterior, outro espelho é adicionado a uma configuração existente do ZFS espelhado.

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

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

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

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

EXEMPLO 3-1 Adicionando discos a uma configuração espelhada do ZFS *(Continuação)*

```

mirror-2  ONLINE      0      0      0
c0t3d0    ONLINE      0      0      0
c1t3d0    ONLINE      0      0      0

```

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

EXEMPLO 3-2 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 pool de armazenamento com um dispositivo RAID-Z que contém três discos em um pool 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 3-3 Adicionando e removendo um dispositivo de log espelhado

O exemplo a seguir mostra como adicionar um dispositivo de log espelhado a um pool de armazenamento espelhado.

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

```
# 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
```

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 pool de armazenamento não espelhado.

É possível remover dispositivos de log usando 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:

    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
```

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

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

Se a configuração do pool contiver apenas um dispositivo de log, remova o dispositivo de log especificando 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 3-4 Adicionando e removendo dispositivos de cache

É possível adicionar dispositivos de cache ao pool de armazenamento do 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
```

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

EXEMPLO 3-4 Adicionando e removendo dispositivos de cache (Continuação)

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 discos de reserva com reposição em funcionamento, dispositivos de log e dispositivos 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 pool.

Para obter mais informações sobre como utilizar dispositivos de cache em um pool de armazenamento ZFS, consulte [“Criando um pool de armazenamento do ZFS com dispositivos de cache” na página 55](#).

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 pool raiz espelhado, consulte [“Como migrar um pool raiz espelhado do ZFS \(pós-instalação\)” na página 120](#).

Se você estiver substituindo um disco em um pool raiz do ZFS, consulte [“Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 166](#).

EXEMPLO 3-5 Convertendo um pool de armazenamento espelhado bidirecional em um pool de armazenamento espalhado tridirecional

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

EXEMPLO 3-5 Convertendo um pool de armazenamento espelhado bidirecional em um pool de armazenamento espalhado tridirecional (Continuação)

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

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

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

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

errors: No known data errors
```

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 3-6 Convertendo um pool de armazenamento não redundante do ZFS em um pool de armazenamento espelhado do ZFS

Além disso, é possível converter um pool de armazenamento não redundante em um pool 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

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

EXEMPLO 3-6 Convertendo um pool de armazenamento não redundante do ZFS em um pool de armazenamento espelhado do ZFS (Continuação)

```
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 pool ao dividir um pool de armazenamento espelhado do ZFS

Um pool de armazenamento espelhado do ZFS pode ser rapidamente clonado como um pool de backup com o comando `zpool split`. Você pode usar esse recurso para dividir um pool raiz espelhado, mas o pool que é dividido não é inicializável até que você execute algumas etapas adicionais.

É possível utilizar o comando `zpool split` para desanexar um ou mais discos de um pool de armazenamento ZFS espelhado para criar um novo pool com o disco ou discos desanexados. O novo pool possuirá conteúdo idêntico ao pool original de armazenamento espelhado do ZFS.

Por padrão, uma operação `zpool split` em um pool espelhado desanexa o último disco do pool recém criado. Após a operação de divisão, importe o novo pool. 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
clt0d0	ONLINE	0	0	0
clt2d0	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
clt0d0	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
clt2d0	ONLINE	0	0	0

errors: No known data errors

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

```
# zpool split tank tank2 clt0d0
```

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 pool e atribuído a um novo GUID de pool. O novo GUID de pool é gerado para que o pool possa ser importado no mesmo sistema no qual foi dividido.

Se o pool a ser dividido possuir pontos de montagem não padrões no sistema de arquivos e o novo pool for criado no mesmo sistema, você deverá utilizar a opção `zpool split -R` para identificar um diretório raiz alternativo para o novo pool 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 pool, importe o novo pool com a opção `-R`. Se o novo pool for criado em um sistema diferente, então não será necessário especificar um diretório raiz alternativo, 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 RAID-Z ou pool não redundante de vários discos.
- Operações de aplicativo e dados devem ser pausados antes de tentar uma operação `zpool split`.
- Um pool não pode ser dividido se o processo de resilvering estiver sendo processado.
- Dividir um pool espelhado é o ideal quando o pool contém dois ou três discos, onde o último disco no pool original é utilizado para o pool recém-criado. Assim, é possível utilizar o comando `zpool attach` para recriar o pool de armazenamento original espelhado ou converter o pool recém-criado em um pool de armazenamento espelhado. No momento não existe um método para criar um pool espelhado *novo* a partir de um pool espelhado *existente* em uma operação `zpool split`, pois o novo pool (divisão) é não redundante
- Se o pool existente for um espelho tridirecional, o novo pool conterá um disco depois da operação de divisão. Se o pool existente for um espelho bidirecional, o resultado será de dois pools não redundantes de dois discos. Você deverá anexar dois discos adicionais para converter os pools não redundantes em pools espelhados.
- Uma boa maneira de manter os dados redundantes durante uma operação de divisão é dividir um pool de armazenamento espelhado que contém três discos para que o pool original contenha dois discos espelhados depois da operação de divisão.
- Confirme se o seu hardware está configurado corretamente antes de dividir um pool espelhado. Para obter informações relacionadas sobre a confirmação do cache do hardware, consulte [“Práticas Gerais do Sistema” na página 321](#).

EXEMPLO 3-7 Dividindo um pool espelhado do ZFS

No exemplo a seguir, um pool de armazenamento espelhado denominado `mothership` com três discos é dividido. Os dois pools resultantes são o pool espelhado `mothership`, com dois discos, e o novo pool, `luna`, com um disco. Cada pool possui conteúdo idêntico.

O pool `luna` pode ser importado em outro sistema para fins de backup. Depois da conclusão do backup, o pool `luna` pode ser destruído e o disco reanexado a `mothership`. Em seguida, é possível repetir o processo.

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

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

EXEMPLO 3-7 Dividindo um pool espelhado do ZFS (Continuação)

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

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

errors: No known data errors

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

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

errors: No known data errors
```

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.

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 c0t5000C500335F95E3d0
```

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 UNAVAIL. 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 c0t5000C500335F95E3d0
cannot offline c0t5000C500335F95E3d0: 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
```

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 pool, mesmo depois de o pool 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 quiser utilizar o dispositivo off-line em outro pool de armazenamento depois de ter destruído o pool de armazenamento original, primeiro coloque o dispositivo on-line, e então destrua o pool de armazenamento original.

Outra forma de utilizar um dispositivo de outro pool de armazenamento, enquanto mantém o pool de armazenamento original, é substituindo o dispositivo existente no pool 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 75](#).

Os dispositivos off-line aparecem no estado OFF-LINE quando é consultado o estado do pool. Para obter informações sobre consultas de status do pool, consulte [“Consultando status de pool de armazenamento do ZFS” na página 86](#).

Para obter mais informações sobre a integridade de dispositivos, consulte [“Determinando o status de integridade de pools de armazenamento do ZFS” na página 92](#).

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 c0t5000C500335F95E3d0
```

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

Se você tentar colocar um dispositivo UNAVAIL.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 danificado exibida no console ou no arquivo /var/adm/messages. Por exemplo:

```
SUNW-MSG-ID: ZFS-8000-D3, TYPE: Fault, VER: 1, SEVERITY: Major
EVENT-TIME: Wed Jun 20 11:35:26 MDT 2012
PLATFORM: SUNW,Sun-Fire-880, CSN: -, HOSTNAME: neo
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: 504a1188-b270-4ab0-af4e-8a77680576b8
DESC: A ZFS device failed. Refer to http://sun.com/msg/ZFS-8000-D3 for more information.
AUTO-RESPONSE: No automated response will occur.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.
```

Para obter mais informações sobre a substituição de um dispositivo danificado, consulte [“Resolvendo um Dispositivo Ausente ou Removido” na página 295](#).

Você pode usar o comando `zpool online -e` para expandir o tamanho do pool se um disco maior foi conectado ao pool ou se um disco menor foi substituído por um disco maior. Por padrão, um disco que é adicionado a um pool não é expandido ao seu tamanho total a não ser que a propriedade do pool `autoexpand` esteja ativada. É possível expandir automaticamente o pool utilizando o comando `zpool online -e`, mesmo se o disco de substituição já estiver on-line ou se o disco estiver atualmente off-line. Por exemplo:

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

Limpando erros de dispositivo do pool 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 c0t5000C500335F95E3d0
```

Para obter mais informações sobre a limpeza de erros de zpool, consulte [“Apagando Erros Transitórios do Dispositivo” na página 300](#).

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 pool redundante, só será preciso identificar o dispositivo substituído. Em alguns hardwares, o ZFS reconhece que o dispositivo é um disco diferente no mesmo local. Por exemplo, para substituir um disco danificado (`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 pool de armazenamento por um disco em uma localização física diferente, você deverá especificar ambos os dispositivos. Por exemplo:

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

Se você estiver substituindo um disco no pool de raiz do ZFS, consulte [“Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 166](#).

A seguir, as etapas básicas para substituir um disco:

1. Coloque o disco off-line, se necessário, com o comando `zpool offline`.
2. Remova o disco que será substituído.
3. Insira o disco de substituição.
4. Examine a saída `format` para determinar se o disco sobressalente está visível.
Além disso, verifique se o ID do dispositivo foi alterado. Se o disco sobressalente tiver um WWN, o ID do dispositivo do disco com falha será alterado.
5. Informe ao ZFS que o disco foi substituído. Por exemplo:

```
# zpool replace tank c1t1d0
```

Se o disco sobressalente tiver um ID de dispositivo diferente do identificado acima, inclua o novo ID do dispositivo.

```
# zpool replace tank c0t5000C500335FC3E7d0 c0t5000C500335BA8C3d0
```

6. Se necessário, coloque o disco on-line outra vez com o comando `zpool online`.

7. Notifique o FMA de que o dispositivo foi substituído.

Na saída `fmadm faulty`, identifique a sequência de caracteres `zfs://pool=name/vdev=guid` na seção `Affects`: e especifique essa sequência de caracteres como um argumento para o comando `fmadm repaired`.

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

Em alguns sistemas com discos SATA, é necessário desconfigurar um 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 obter um exemplo de substituição do disco SATA, consulte o [Exemplo 10-1](#).

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

- Se a propriedade `autoreplace` do pool for definida como `on`, qualquer dispositivo novo encontrado na mesma localização física que um dispositivo que antes pertencia ao pool 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 estado do pool de armazenamento `REMOVED` é fornecido quando um dispositivo ou disco de reserva com reposição em funcionamento tiver sido fisicamente removido enquanto o sistema estava em execução. Um disco de reserva com reposição em funcionamento é substituído pelo dispositivo removido, se disponível.
- Se um dispositivo for removido e depois inserido, o dispositivo será colocado on-line. Se um disco de reserva com reposição em funcionamento tiver sido ativado quando o dispositivo foi reinserido, o disco de reserva será 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 discos de reserva com reposição em funcionamento são verificados periodicamente para garantir que estejam on-line e disponíveis.
- O tamanho do dispositivo de substituição deve ser igual ou maior que o menor disco em um uma configuração espelhada ou RAID-Z.

- Quando um dispositivo de substituição com tamanho maior que o dispositivo que está substituindo é adicionado ao pool, ele não é expandido automaticamente para seu tamanho total. A propriedade do pool `autoexpand` determina se o pool é expandido quando um disco maior é adicionado ao pool. Por padrão, a propriedade `autoexpand` é desativada. É possível ativar esta propriedade ao expandir o tamanho do pool antes ou depois de adicionar o maior disco ao pool.

No exemplo a seguir, dois discos de 16 GB em um pool espelhado são substituídos por dois discos de 72 GB. Verifique se o primeiro dispositivo foi completamente polido antes de tentar a substituição do segundo dispositivo. A propriedade `autoexpand` é ativada após as substituições do disco para expandir o tamanho completo do disco.

```
# 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 pool grande é demorado devido ao processo de 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 danificado foi substituído automaticamente por um disco de reserva com reposição em funcionamento, poderá ser necessário desanexar o disco de reserva depois que o disco danificado for substituído. Você pode usar o comando `zpool detach` para desanexar um disco de reserva em um pool espelhado ou RAID-Z. Para obter mais informações sobre a desanexação de um disco de reserva com reposição em funcionamento, consulte [“Ativando e desativando discos de reserva com reposição em funcionamento no pool de armazenamento”](#) na página 80.

Para obter mais informações sobre a substituição de dispositivos, consulte [“Resolvendo um Dispositivo Ausente ou Removido”](#) na página 295 e [“Substituindo ou reparando um dispositivo modificado”](#) na página 298.

Designando discos de reserva com reposição em funcionamento no pool de armazenamento

O recurso de discos de reserva com reposição em funcionamento permite identificar os discos que poderiam ser usados para substituir um dispositivo danificado em um pool de armazenamento. A atribuição de um dispositivo como *reserva* significa que o dispositivo não é um dispositivo ativo em um pool, mas se um dispositivo ativo no pool falhar, o de reserva substituirá automaticamente o dispositivo danificado.

Os dispositivos podem ser designados como reserva nas seguintes situações:

- Quando o pool é criado com o comando `zpool create`.
- Após o pool ser criado com o comando `zpool add`.

Os seguintes exemplos mostram como atribuir dispositivos como discos de reserva com reposição em funcionamento quando o pool é criado:

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

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

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

O seguinte exemplo mostra como atribuir discos de reserva com reposição em funcionamento ao adicioná-los a um pool depois que o pool é criado:

```
# zpool add zeepool spare c0t5000C500335E106Bd0 c0t5000C500335FC3E7d0
# zpool status zeepool
pool: zeepool
```

```
state: ONLINE
scan: none requested
config:
```

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

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

Os discos de reserva com reposição em funcionamento podem ser removidos de um pool de armazenamento com o comando `zpool remove`. Por exemplo:

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

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

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

Um disco de reserva com reposição em funcionamento não pode ser removido se estiver sendo utilizado atualmente por um pool de armazenamento.

Considere o seguinte quando estiver usando discos de reserva com reposição em funcionamento do ZFS:

- Atualmente, o comando `zpool remove` pode ser utilizado apenas para remover discos de reserva com reposição em funcionamento, dispositivos de cache e dispositivos de log.

- Para adicionar um disco como reserva, o disco de reserva deve ter tamanho igual ou maior ao do maior disco no pool. É permitido adicionar um disco menor como um disco de reserva de um pool. No entanto, quando o disco de reserva 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 discos de reserva com reposição em funcionamento no pool de armazenamento

Os discos de reserva com reposição em funcionamento são ativados das seguintes maneiras:

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

Se um disco de reserva atualmente em uso falhar, o agente desanexará esse disco de reserva, cancelando a substituição. Em seguida, o agente tentará substituir o dispositivo por outro disco de reserva, se houver um disponível. Esse recurso é atualmente limitado pelo fato de o mecanismo de diagnóstico do ZFS emitir falhas somente quando um dispositivo desaparece do sistema.

Se você substituir fisicamente um dispositivo danificado por um disco de reserva ativo, poderá reativar o dispositivo original, utilizando o comando `zpool detach` para desanexar o disco de reserva. Se definir a propriedade `autoreplace` do pool como `on`, o disco de reserva será desanexado e devolvido automaticamente para o pool disponível quando o novo dispositivo for inserido e a operação on-line terminar.

Um dispositivo UNAVAIL é substituído automaticamente quando um disco de reserva está disponível. Por exemplo:

```
# zpool status -x
pool: zeepool
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scan: resilvered 3.15G in 0h0m with 0 errors on Mon Nov 12 15:53:42 2012
config:
```

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


```

mirror-1          DEGRADED      0      0      0
c0t5000C500335BD117d0  ONLINE      0      0      0
spare-1           DEGRADED    449      0      0
c0t5000C500335DC60Fd0  UNAVAIL      0      0      0
c0t5000C500335E106Bd0  ONLINE      0      0      0
spares
c0t5000C500335E106Bd0  INUSE

```

errors: No known data errors

Atualmente, é possível desativar um disco de reserva das seguintes maneiras:

- Ao remover o disco de reserva do pool de armazenamento.
- Ao desanexar um disco de reserva depois que o disco danificado é substituído fisicamente. Consulte [Exemplo 3–8](#).
- Ao permutar temporária ou permanentemente outro disco de reserva. Consulte [Exemplo 3–9](#).

EXEMPLO 3–8 Desanexando um disco de reserva depois que o disco danificado é substituído

Neste exemplo, o disco danificado (c0t5000C500335DC60Fd0) é substituído fisicamente e o ZFS é notificado ao utilizar o comando `zpool replace`.

```

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

```

```

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

```

Se necessário, você pode utilizar o comando `zpool detach` para retornar o disco de reserva para o pool disponível. Por exemplo:

```
# zpool detach zeepool c0t5000C500335E106Bd0
```

EXEMPLO 3–9 Desanexando um disco danificado e utilizando o disco de reserva

Se quiser substituir um disco danificado, fazendo uma permuta temporária ou permanente do disco de reserva que está atualmente substituindo-o, desanexe o disco original (com falha). Se o disco danificado é substituído eventualmente, você poderá adicioná-lo de volta ao pool de armazenamento como um disco de reserva. Por exemplo:

EXEMPLO 3-9 Desanexando um disco danificado e utilizando o disco de reserva (Continuação)

```
# zpool status zeepool
pool: zeepool
state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scan: resilver in progress since Mon Nov 12 16:04:12 2012
      4.80G scanned out of 12.0G at 55.8M/s, 0h2m to go
      4.80G scanned out of 12.0G at 55.8M/s, 0h2m to go
      4.77G resilvered, 39.97% done
config:
```

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

```
errors: No known data errors
# zpool detach zeepool c0t5000C500335DC60Fd0
# zpool status zeepool
pool: zeepool
state: ONLINE
scan: resilvered 11.3G in 0h3m with 0 errors on Mon Nov 12 16:07:12 2012
config:
```

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

```
errors: No known data errors
(Original failed disk c0t5000C500335DC60Fd0 is physically replaced)
# zpool add zeepool spare c0t5000C500335DC60Fd0
# zpool status zeepool
pool: zeepool
state: ONLINE
scan: resilvered 11.2G in 0h3m with 0 errors on Mon Nov 12 16:07:12 2012
config:
```

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

EXEMPLO 3-9 Desanexando um disco danificado e utilizando o disco de reserva (Continuação)

```

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

```

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

Depois que um disco é substituído e o disco de reserva é desanexado, informe ao FMA que o disco foi reparado.

```

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

```

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 tank
tank size          68G          -
tank capacity      0%           -
tank altroot       -            default
tank health        ONLINE       -
tank guid          15560293364730146756 -
tank version       32           default
tank bootfs        -            default
tank delegation    on           default
tank autoreplace   off          default
tank cachefile     -            default
tank failmode      wait         default
tank listsnapshots on           default
tank autoexpand    off          default
tank free          68.0G        -
tank allocated     124K         -
tank readonly      off          -

```

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

```

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

```

Se você tentar definir uma propriedade em um pool que esteja 100% cheio, verá uma mensagem semelhante a esta:

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

Para obter informações sobre como impedir problemas de capacidade de espaço do pool, consulte o [Capítulo 11, “Práticas Recomendadas do Oracle Solaris ZFS”](#).

TABELA 3-1 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 pool 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 pool 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	Booliano	off	Controla a substituição automática do dispositivo. Se for definido como desativado, a substituição do dispositivo deve ser iniciada pelo administrador utilizando o comando <code>zpool replace</code> . Se for definido como ativado, qualquer dispositivo novo encontrado no mesmo local físico que um dispositivo que antes pertencia ao pool será automaticamente formatado e substituído. A abreviação da propriedade é <code>replace</code> .
bootfs	Booliano	N/D	Identifica o sistema de arquivos inicializável padrão do pool raiz. Esta propriedade é tipicamente definida pelos programas de instalação.
cachefile	Sequência de caracteres	N/D	Controla onde as informações de configuração do pool são armazenadas. Todos os pools 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 pools não sejam automaticamente importados. É possível definir esta propriedade para armazenar as informações de configuração do pool 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 do ZFS, esta propriedade não é utilizada.

TABELA 3-1 Descrições das propriedades do pool do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
capacidade	Número	N/D	Valor somente leitura que identifica a porcentagem de espaço usado do pool. A abreviatura da propriedade é <code>cap</code> .
delegação	Booleano	on	Controla se um usuário não privilegiado pode obter permissões de acesso definidas para o sistema de arquivos. Para mais informações, consulte o Capítulo 8, “Administração delegada do ZFS do Oracle Solaris” .
failmode	Sequência de caracteres	espera	Controla o comportamento do sistema se uma falha catastrófica do pool 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 pool. O comportamento de tal evento é determinado por um dos seguintes valores: <ul style="list-style-type: none"> ■ espera: bloqueia todos os acessos de E/S ao pool até que a conectividade com o dispositivo seja restaurada e os erros sejam apagados com o comando <code>zpool clear</code>. Neste estado, operações de E/S para o pool são bloqueadas, mas operações de leitura podem ter êxito. Um pool é mantido no estado <code>wait</code> até que o problema do dispositivo seja resolvido. ■ continuar: 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>. ■ pânico: imprime uma mensagem ao console e gera um despejo de memória de sistema.
livre	Sequência de caracteres	N/D	Valor de somente leitura que identifica o número de bloqueios dentro do pool que não é alocado.
guid	Sequência de caracteres	N/D	Valor somente leitura que identifica o identificador exclusivo do pool.
integridade	Sequência de caracteres	N/D	Valor somente leitura que identifica a integridade atual do pool, como ONLINE, DEGRADED, SUSPENDED, REMOVED ou UNAVAIL. propriedade

TABELA 3-1 Descrições das propriedades do pool do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
listshares	Sequência de caracteres	off	Controla se as informações de compartilhamento deste pool serão exibidas com o comando <code>zfs list</code> . O valor padrão é <code>off</code> .
listsnapshots	Sequência de caracteres	on	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> .
tamanho	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 pool. 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 pools de armazenamento do ZFS” na página 86
- “Visualizando estatísticas de E/S para pools de armazenamento do ZFS” na página 90
- “Determinando o status de integridade de pools de armazenamento do ZFS” na página 92

Exibindo informações sobre os pools de armazenamento do ZFS

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

Exibindo informações sobre todos os pools de armazenamento ou de um pool específico

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

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

Esta saída de comando exibe as seguintes informações:

NAME	O nome do pool.
SIZE	O tamanho total do pool, igual à soma do tamanho de todos os dispositivos virtuais de nível superior.
ALLOC	O 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 32.
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 92.
ALTROOT	A raiz alternativa do pool, se houver alguma. Para obter mais informações sobre pools de raiz alternativa, consulte “Usando pools de raiz alternativa do ZFS” na página 284.

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  -
```

Você pode usar o intervalo `zpool list` e as opções de contagem para reunir estatísticas sobre um período. Além disso, você pode exibir um registro de data e hora usando a opção `-T`. Por exemplo:

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

```
rpool 33.8G 12.2G 21.5G 36% 1.00x ONLINE -
Tue Nov 2 10:36:14 MDT 2010
pool 33.8G 83.5K 33.7G 0% 1.00x ONLINE -
rpool 33.8G 12.2G 21.5G 36% 1.00x ONLINE -
```

Exibindo 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 [“Exibindo informações sobre todos os pools de armazenamento ou de um pool específico” na página 86](#).

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 pools no sistema, utilize a seguinte sintaxe:

```
# zpool list -H -o 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 clt0d0s0 clt1d0s0
```



```

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 pools menores, o tamanho máximo está limitado a 1% do tamanho do pool, nos quais o *tamanho* é determinado no momento da criação do pool.
- 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
2012-01-25.16:35:32 zpool create -f tank mirror c3t1d0 c3t2d0 spare c3t3d0
2012-02-17.13:04:10 zfs create tank/test
2012-02-17.13:05:01 zfs snapshot -r tank/test@snap1

```

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

```

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

```

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

Visualizando estatísticas de E/S para pools de armazenamento do 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:

capacidade de alocação	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 o espaço do pool e o espaço do conjunto de dados, consulte “Contabilidade de espaço em disco do ZFS” na página 32 .
capacidade livre	A quantidade de espaço de disco disponível no pool 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.
operações de leitura	O número de operações de E/S de leitura enviado para o pool ou dispositivo, incluindo solicitações de metadados.
operações de gravação	O número de operações de E/S de gravação enviado para o pool ou dispositivo.
ler largura de banda	A largura de banda de todas as operações de leitura (incluindo metadados), expressa como unidades por segundo.

largura de banda para gravação

A largura de banda de todas as operações de gravação, expressa como unidades por segundo.

Listando estatísticas de E/S em todo o pool

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
```

No exemplo acima, o comando exibe estatísticas de uso do pool `tank` a cada dois segundos até você digitar Control-C. Como alternativa, você pode especificar um argumento `count` adicional, que causa o encerramento do comando 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 pool, as estatísticas serão exibidas em linhas consecutivas. Se houver mais de um pool, uma linha tracejada adicional delineará cada iteração para fornecer separação visual.

Listando estatísticas de E/S do dispositivo virtual

Além das estatísticas de E/S de todos os pool, 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
clt0d0s0	-	-	0	0	578	109
clt1d0s0	-	-	0	0	595	109
tank	36.5G	31.5G	4	1	295K	146K
mirror	36.5G	31.5G	126	45	8.13M	4.01M
clt2d0	-	-	0	3	100K	386K
clt3d0	-	-	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 pool é criado, uma vez que uma quantidade significativa de E/S é feita diretamente para os discos como parte da criação do pool, 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 pool são relatadas pelo `fmd` 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 obter mais informações sobre a solução de problemas e a recuperação de dados, consulte [Capítulo 10, “Solução de problemas e recuperação de pools do Oracle Solaris ZFS”](#).

O status de integridade de um pool é descrito por um destes quatro estados:

DEGRADED

Um pool com um ou mais dispositivos com falha, mas os dados permanecem disponíveis devido a uma configuração redundante.

ONLINE

Um pool que tem todos os dispositivos em operação normal.

SUSPENDED

Um pool que está esperando a conectividade de um dispositivo para ser restaurado. Um pool **SUSPENDED** permanece no estado de espera até que o problema do dispositivo seja resolvido.

UNAVAIL

Um pool com metadados corrompidos, um ou mais dispositivos indisponíveis e réplicas insuficientes para continuar funcionando.

Cada pool pode se enquadrar em um dos seguintes estados:

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.
OFFLINE	O dispositivo foi colocado explicitamente off-line pelo administrador.
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.
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.
UNAVAIL	O dispositivo ou o dispositivo virtual não pode ser aberto. Em alguns casos, pools com dispositivos UNAVAIL aparecem no modo DEGRADED . Se um dispositivo virtual de nível superior estiver UNAVAIL , nada no pool poderá ser acessado.

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 pool também estará **DEGRADED**. Se um dispositivo virtual de primeiro nível estiver **UNAVAIL** ou **OFFLINE**, o pool também estará **UNAVAIL** ou **SUSPENDED**. Um pool no estado **UNAVAIL** ou **SUSPENDED** fica totalmente inacessível. Nenhum dado poderá ser recuperado até que os dispositivos necessários sejam anexados ou reparados. Um pool no estado **DEGRADED** continua a ser executado, mas talvez não atinja o mesmo nível de redundância de dados ou throughput de dados se o pool estivesse on-line.

O comando `zpool status` também fornece detalhes sobre operações de resilver e scrub.

- Relatório de andamento da operação de resilver. Por exemplo:

```
scan: resilver in progress since Wed Jun 20 14:19:38 2012
      7.43G scanned out of 71.8G at 36.4M/s, 0h30m to go
      7.43G resilvered, 10.35% done
```

- Relatório de andamento da operação de scrub. Por exemplo:

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

- Mensagem de conclusão da operação de resilver. Por exemplo:

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

- Mensagem de conclusão da operação de scrub. Por exemplo:

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

- Mensagem de cancelamento da operação de scrub em andamento. Por exemplo:

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

- Mensagens de conclusão de scrub e de resilver persistem nas reinicializações do sistema

Status básico de integridade do pool de armazenamento

É possível rever rapidamente o estado de integridade dos pools ao utilizar o comando `zpool status` desta maneira:

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

É possível examinar um determinado pool ao especificar o nome de um pool 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	
clt0d0	ONLINE	0	0	0	
clt1d0	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 pool.

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

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

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

A saída acima identifica que o dispositivo permanece em um estado com falhas até a conclusão de uma operação de resilver.

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

Se um pool tiver um dispositivo off-line, a saída do comando identificará o pool 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

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

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

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

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 pools e dados UNAVAIL, consulte o [Capítulo 10, “Solução de problemas e recuperação de pools do Oracle Solaris ZFS”](#).

Reunindo informações do status do pool de armazenamento do ZFS

Você pode usar o intervalo `zpool status` e as opções de contagem para reunir estatísticas sobre um período. Além disso, você pode exibir um registro de data e hora usando a opção `-T`. Por exemplo:

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

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


```

c0t5000C500335BD117d0 ONLINE      0      0      0
c0t5000C500335DC60Fd0 ONLINE      0      0      0

errors: No known data errors

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

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

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

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

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

errors: No known data errors

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

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

errors: No known data errors

```

Migrando pools de armazenamento do ZFS

Ocasionalmente, pode ser necessário mover um pool 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 pool de um sistema e importá-lo para o sistema de destino, mesmo que os sistemas tenham arquitetura de endianness diferentes. Para obter informações sobre replicação ou migração de sistemas de arquivos entre diferentes pools de armazenamento que possam residir em diferentes sistemas, consulte [“Enviando e recebendo dados do ZFS” na página 223](#).

- [“Preparando para migração do pool de armazenamento do ZFS” na página 98](#)
- [“Exportando um pool de armazenamento do ZFS” na página 98](#)
- [“Determinando pools de armazenamento disponíveis para importação” na página 99](#)
- [“Importando pools de armazenamento ZFS de diretórios alternativos” na página 101](#)
- [“Importando pools de armazenamento do ZFS” na página 101](#)
- [“Recuperando pools de armazenamento destruídos do ZFS” na página 105](#)

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 pool.

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 pool aparecerá como UNAVAIL no sistema original porque os dispositivos não estão mais presentes. Por padrão, o sistema de destino não pode importar um pool que não tenha sido exportado explicitamente. Essa condição é necessária para impedir a importação acidental de um pool 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/eric': Device busy
# zpool export -f tank
```

Depois que o comando é executado, o pool 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 pool 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 275](#).

Determinando pools de armazenamento disponíveis para importação

Depois que o pool 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 pool 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 pools 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 pool 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 pool, mas houver redundância disponível suficiente para fornecer um pool utilizável, o pool 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 indisponíveis, o pool não poderá ser importado.

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 poderá determinar quais eram os outros dispositivos que faziam parte do pool, 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 pools 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 você estiver utilizando pools baseados em 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 pool é identificado para importação, é possível importá-lo especificando o nome do pool ou o seu identificador numérico como um argumento para o comando `zpool import`. Por exemplo:

```
# zpool import tank
```

Se vários pools disponíveis tiverem o mesmo nome, especifique o pool 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
c1t8d0     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. O nome do novo pool é persistente.

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
```

Observação – Não tente importar um pool que está ativo em um sistema para outro sistema. O ZFS não é um sistema de arquivos cluster nativo, distribuído ou paralelo, e não pode fornecer acesso de vários host diferentes.

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 284](#).

Importação de um pool com um dispositivo de log ausente

Por padrão, um pool com um dispositivo de log ausente não pode ser importado. Você pode usar o comando `zpool import -m` para forçar um pool a ser importado com um dispositivo de log ausente. Por exemplo:

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

cannot import 'dozer': one or more devices is currently unavailable
```

Importe o pool com o dispositivo de log ausente. Por exemplo:

```
# zpool import -m dozer
# zpool status dozer
pool: dozer
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scan: none requested
config:
```

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

errors: No known data errors

Depois de conectar o dispositivo de log ausente, execute o comando `zpool clear` para limpar os erros do pool.

Uma recuperação semelhante pode ser tentada com dispositivos de log espelhados ausentes. Por exemplo:

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

cannot import 'dozer': one or more devices is currently unavailable

```
# zpool import -m dozer
# zpool status dozer
pool: dozer
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scan: scrub repaired 0 in 0h0m with 0 errors on Fri Oct 15 16:51:39 2010
config:
```

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

Depois de conectar os dispositivos de log ausentes, execute o comando `zpool clear` para limpar os erros do pool.

Importação de um pool no modo somente leitura

Você pode importar um pool no modo somente leitura. Se um pool for danificado de forma que não possa ser acessado, esse recurso pode permitir a recuperação dos dados do pool. Por exemplo:

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

Quando um pool é importado no modo somente leitura, as seguintes condições se aplicam:

- Todos os sistemas e volumes são montados no modo somente leitura.
- O processamento de transações do pool é desativado. Isso também significa que qualquer gravação síncrona pendente no log de intenções não será reproduzida até que o pool seja importado no modo leitura-gravação.
- As tentativas de definir uma propriedade do pool durante a importação somente leitura são ignoradas.

Um pool somente leitura pode ser definido de volta para o modo leitura-gravação exportando e importando o pool. Por exemplo:

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

Importação de um pool por um caminho de dispositivo específico

O seguinte comando importa o pool `dpool` identificando um dos dispositivos específicos do pool, `/dev/dsk/c2t3d0`, neste exemplo.

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

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

Embora este pool esteja composto por discos inteiros, o comando deve incluir o identificador de parte do dispositivo específico.

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 pool `tank` como o pool destruído por causa das seguintes informações de estado:

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

Para recuperar o pool destruído, execute o comando `zpool import -D` novamente com o pool 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 não estiver disponível, você poderá recuperar o pool destruído mesmo assim, ao incluir a opção `-f`. Neste cenário, importe o pool degradado e, em seguida, tente corrigir a falha do dispositivo. Por exemplo:

```
# zpool destroy dozer
# zpool import -D
  pool: dozer
    id: 4107023015970708695
  state: DEGRADED (DESTROYED)
 status: One or more devices could not be opened.  Sufficient replicas exist for
```

```
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
config:
    dozer                DEGRADED
    raidz2-0             DEGRADED
    c8t0d0               ONLINE
    c8t1d0               ONLINE
    c8t2d0               ONLINE
    c8t3d0               UNAVAIL  cannot open
    c8t4d0               ONLINE
errors: No known data errors
# zpool import -Df dozer
# zpool status -x
    pool: dozer
    state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-2Q
scan: none requested
config:

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

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

Atualizando pools de armazenamento do ZFS

Se os pools de armazenamento do ZFS forem de uma versão anterior do Solaris, será possível atualizar os pools com o comando `zpool upgrade` para tirar proveito dos recursos de pool da versão atual. Além disso, o comando `zpool status` notifica 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 refreservation properties
10   Cache devices
11   Improved scrub performance
12   Snapshot properties
13   snapused property
14   passthrough-x aclinherit
15   user/group space accounting
16   stmf property support
17   Triple-parity RAID-Z
18   Snapshot user holds
19   Log device removal
20   Compression using zle (zero-length encoding)
21   Reserved
22   Received properties

```

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

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.

Se desejar utilizar o console de administração do ZFS em um sistema com um pool de uma versão anterior do Solaris, certifique-se de atualizar os pools antes de utilizar o console. Para determinar se os pools precisam ser atualizados, utilize o comando `zpool status`.

Instalando e inicializando um sistema de arquivos raiz do Oracle Solaris ZFS

Este capítulo descreve como instalar e inicializar um sistema de arquivos raiz do Oracle Solaris ZFS. Também é tratada a migração de um sistema de arquivos raiz UFS para um sistema de arquivos ZFS com o recurso Oracle Solaris Live Upgrade.

Este capítulo traz as seguintes seções:

- “Instalando e inicializando um sistema de arquivos do Oracle Solaris ZFS (visão geral)” na página 109
- “Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS” na página 111
- “Instalação de um sistema de arquivos raiz ZFS (instalação inicial do Oracle Solaris)” na página 114
- “Como migrar um pool raiz espelhado do ZFS (pós-instalação)” na página 120
- “Instalando um sistema de arquivos raiz ZFS (instalação de arquivo do Oracle Solaris Flash)” na página 121
- “Instalando um sistema de arquivos raiz ZFS (Instalação com JumpStart)” na página 126
- “Migrando um Sistema de Arquivos de Raiz do ZFS ou Atualizando um Sistema de Arquivos de Raiz do ZFS (Live Upgrade)” na página 129
- “Gerenciamento dos dispositivos de permuta e despejo do ZFS” na página 154
- “Inicializando a partir de um sistema de arquivos raiz ZFS” na página 159
- “Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 166

Para uma lista de problemas conhecidos nesta versão, consulte *Notas da versão Oracle Solaris 10 1/13*.

Instalando e inicializando um sistema de arquivos do Oracle Solaris ZFS (visão geral)

É possível instalar e inicializar a partir de um sistema de arquivos raiz ZFS das seguintes formas:

- **Instalação inicial do Oracle Solaris (método de instalação no modo de texto interativo)**
 - Selecione e instale o ZFS como o sistema de arquivos raiz.

- Instale um arquivo flash do ZFS.
- **Recurso Oracle Solaris Live Upgrade**
 - Migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS.
 - Criar um novo ambiente de inicialização em um novo pool raiz do ZFS.
 - Crie ou atualize um ambiente de inicialização em um pool raiz do ZFS existente.
 - Atualize um ambiente de inicialização (BE) alternativo com um arquivo flash do ZFS.
- **Recurso Oracle Solaris JumpStart**
 - Crie um perfil para instalar automaticamente um sistema de arquivos raiz ZFS.
 - Crie um perfil para instalar automaticamente 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 159](#).

Recursos de instalação do ZFS

São oferecidos os seguintes recursos de instalação do ZFS nesta versão do Oracle Solaris:

- Com o recurso do instalador de texto interativo, é possível instalar um sistema de arquivos raiz UFS ou ZFS. O sistema de arquivos padrão ainda é o UFS nesta versão do Oracle Solaris. Você pode acessar o instalador de texto interativo das seguintes maneiras:
 - SPARC: utilize a seguinte sintaxe do DVD de instalação do Oracle Solaris:

```
ok boot cdrom - text
```
 - SPARC: utilize a seguinte sintaxe quando for inicializar a partir da rede:

```
ok boot net - text
```
 - x86: selecione o método de instalação no modo de texto.
- Um perfil JumpStart personalizado oferece os seguintes recursos:
 - 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 instalar um arquivo flash de um pool raiz do ZFS.
- Utilizando o Live Upgrade, é possível migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS.
- É possível configurar um pool raiz espelhado do ZFS selecionando dois discos durante a instalação. Ou, é possível anexar discos adicionais depois da instalação para criar um pool raiz espelhado do ZFS.
- Os dispositivos de permuta e despejo são criados automaticamente nos volumes ZFS do pool raiz do 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. Você deve selecionar o método de instalação no modo de texto para instalar um sistema de arquivos raiz ZFS.
- 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 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 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 raiz do ZFS

As seções a seguir descrevem o espaço e os requisitos de configuração do pool raiz do ZFS.

Requisitos de espaço em disco para pools raiz do ZFS

O espaço mínimo necessário 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 permuta e despejo devem ser dispositivos separados em um ambiente raiz do ZFS. Por padrão, os dispositivos de permuta e despejo são os mesmos 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 permuta e o dispositivo de despejo dependem da quantidade de memória física. O espaço mínimo 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.

Verifique os seguintes requisitos de espaço em disco para pools de armazenamento do ZFS:

- 1536 MB é a quantidade de memória mínima necessária para instalar um sistema de arquivos raiz ZFS.

- 1536 MB de memória ou mais é recomendado para melhor desempenho geral do ZFS.
- Pelo menos 16 GB de espaço em disco são recomendados. O espaço de disco é consumido da seguinte forma:
 - **Área de permuta e dispositivo de despejo** – Os tamanhos padrão dos volumes de permuta e despejo criados por programas de instalação do Oracle Solaris são:
 - **Instalação inicial** – No novo ambiente de inicialização ZFS, o tamanho 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 de despejo 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.
 - **Live Upgrade** – quando um sistema de arquivos raiz UFS é migrado para um sistema de arquivos raiz ZFS, o tamanho padrão do 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 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 de permuta é definido em 512 MB.
 - No BE do ZFS, o tamanho do 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 156.](#)

- **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 no mesmo pool raiz usam os mesmos dispositivos de permuta e despejo.
- **Componentes do SO Oracle Solaris:** 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 SO devem residir no pool raiz, com exceção dos dispositivos de permuta e despejo.

Para obter informações sobre alteração da permuta padrão e dos dispositivos de despejo com o Live Upgrade, consulte [“Personalizando Volumes de Permuta e Dump do ZFS” na página 157.](#)

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 Live Upgrade para migrar ou instalar patches em um BE do ZFS ou criar um arquivo Flash ZFS deste pool.

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 de configuração do pool raiz do ZFS

Verifique os seguintes requisitos de configuração do pool raiz do ZFS:

- O pool que foi designado para ser o pool raiz precisa ter um rótulo SMI. Este requisito geralmente é atendido caso o pool seja criado com segmentos de disco.
- O pool deve existir tanto em um segmento de disco quanto em segmentos de disco espelhados. Se você tentar usar uma configuração de pool não suportada durante a migração do Live Upgrade, verá 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 do ZFS, consulte [“Criando um pool raiz do ZFS” na página 51](#).

- x86: o disco deve conter uma partição fdisk do Oracle Solaris. Esta partição fdisk é criada automaticamente quando o sistema baseado em 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 do ZFS devem ser menores que 2 TB em sistemas baseados em SPARC e em x86.
- A compactação pode ser habilitada no pool raiz, mas somente após o pool raiz ser instalado. Não há como habilitar a compactação em um pool raiz durante a instalação. O algoritmo de compactação gzip não é suportado no pool raiz.
- Não renomeie o pool raiz depois de ele ter sido criado por uma instalação inicial ou depois da migração do Solaris Live Upgrade para um sistema de arquivos raiz do ZFS. Renomear o pool raiz pode resultar em um sistema não inicializável.

Além disso, não altere o ponto de montagem padrão dos componentes do pool raiz, se quiser usar o Live Upgrade.

- Se você deseja alterar os dispositivos de permuta e despejo e usar o Live Upgrade, consulte [“Personalizando Volumes de Permuta e Dump do ZFS” na página 157](#).

Instalação de um sistema de arquivos raiz ZFS (instalação inicial do Oracle Solaris)

Nesta versão do Oracle Solaris, você pode executar uma instalação inicial usando os métodos a seguir:

- Use o instalador de texto interativo para instalar inicialmente um pool de armazenamento do ZFS que contenha um sistema de arquivos raiz ZFS inicializável. Se tiver um pool de armazenamento do ZFS que deseja usar no sistema de arquivos raiz ZFS, você deve usar o Live Upgrade para migrar o sistema de arquivos raiz UFS existente em um pool de armazenamento do ZFS existente. Para obter mais informações, consulte [“Migrando um Sistema de Arquivos de Raiz do ZFS ou Atualizando um Sistema de Arquivos de Raiz do ZFS \(Live Upgrade\)”](#) na página 129.
- Use o instalador de texto interativo para instalar inicialmente um pool de armazenamento do ZFS que contenha um sistema de arquivos raiz ZFS inicializável de um arquivo flash ZFS.

Antes de começar a instalação inicial para criar um pool de armazenamento do ZFS, consulte [“Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS”](#) na página 111.

Se for configurar zonas depois da instalação inicial de um sistema de arquivos raiz ZFS e tiver planos de atualizar ou instalar patches no sistema, consulte [“Usando o Live Upgrade para migrar ou atualizar um sistema com zonas \(Solaris 10 10/08\)”](#) na página 139 ou [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas \(no mínimo Solaris 10 5/09\)”](#) na página 144.

Se já possuir pools de armazenamento do ZFS no sistema, eles serão reconhecidos pela mensagem a seguir. Porém, esses pools permanecem intocados, a menos que você selecione os discos nos pools existentes para criar um novo pool 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 estiver selecionado no novo pool.

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável

O processo de instalação de texto interativo é basicamente o mesmo das versões anteriores do Oracle Solaris, exceto que é solicitado que você crie um sistema de arquivos raiz UFS ou ZFS. O UFS ainda é o sistema de arquivos padrão nesta versão. Se você selecionar um sistema de arquivos raiz ZFS, será solicitada a criação de um pool de armazenamento do ZFS. As etapas para instalar o sistema de arquivos raiz ZFS seguem:

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

1. Insira a mídia de instalação do Oracle Solaris ou inicialize o sistema a partir de um servidor de instalação. Em seguida, selecione o método de instalação de texto interativo para criar um sistema de arquivos raiz ZFS inicializável.

- SPARC: utilize a sintaxe a seguir do DVD de instalação do Oracle Solaris:

```
ok boot cdrom - text
```

- SPARC: utilize a sintaxe a seguir quando for inicializar a partir da rede:

```
ok boot net - text
```

- x86: selecione o método de instalação no modo de texto.

Você também pode criar um arquivo flash ZFS a ser instalado usando os métodos a seguir:

- instalação JumpStart. Para obter mais informações, consulte o [Exemplo 4-2](#).
- Instalação inicial. Para obter mais informações, consulte o [Exemplo 4-3](#).

Você pode executar uma atualização padrão para atualizar um sistema de arquivos ZFS inicializável existente, mas não pode usar esta opção para criar um novo sistema de arquivos ZFS inicializável. 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 de Raiz do ZFS ou Atualizando um Sistema de Arquivos de Raiz do ZFS \(Live Upgrade\)](#)” na página 129.

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 do ZFS. Esta tela é semelhante à das versões anteriores do

```
Select Disks
```

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

Disk Device	Available Space
[X] ** c1t0d0	139989 MB (F4 to edit)
) [] c1t1d0	139989 MB
[] c1t2d0	139989 MB

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

```
[ ] c1t3d0 139989 MB
[ ] c2t0d0 139989 MB
[ ] c2t1d0 139989 MB
[ ] c2t2d0 139989 MB
[ ] c2t3d0 139989 MB
Maximum Root Size: 139989 MB
Suggested Minimum: 11102 MB
```

Você pode selecionar um ou mais discos a serem usados para o pool raiz do ZFS. Se selecionar dois discos, uma configuração espelhada de dois discos é definida para o pool raiz. Um pool espelhado de dois ou três discos é o ideal. Se você tiver oito discos e selecionar todos eles, esses oito discos serão utilizados para o pool raiz como um grande espelho. Esta configuração não é considerada ótima. Outra opção é criar um pool raiz espelhado 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 47](#).

- 4. Para selecionar dois discos para criar um pool raiz espelhado, utilize as teclas de controle do cursor para selecionar o segundo disco.

No exemplo a seguir, ambos c0t1d0 e c0t1d0 são selecionados para os discos de pool 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 139989 MB (F4 to edit)
) [X] c1t1d0 139989 MB
[ ] c1t2d0 139989 MB
[ ] c1t3d0 139989 MB
[ ] c2t0d0 139989 MB
[ ] c2t1d0 139989 MB
[ ] c2t2d0 139989 MB
[ ] c2t3d0 139989 MB
Maximum Root Size: 139989 MB
Suggested Minimum: 11102 MB
```

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

Se a coluna Espaço disponível identifica 0 MB, o disco provavelmente possui uma legenda EFI. Se deseja utilizar um disco com um rótulo EFI, é necessário sair do programa de instalação, rotular novamente o disco com um rótulo SMI usando o comando `format -e`, então, reiniciar o programa de instalação.

Se não for criado um pool raiz espelhado durante a instalação, será possível criar um facilmente depois da instalação. Para obter informações, consulte [“Como migrar um pool raiz espelhado do ZFS \(pós-instalação\)” na página 120](#).

Depois de selecionar um ou mais discos para o pool de armazenamento do 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
        ZFS Root Dataset Name: sl0nameBE
        ZFS Pool Size (in MB): 139990
        Size of Swap Area (in MB): 4096
        Size of Dump Area (in MB): 1024
        (Pool size must be between 7006 MB and 139990 MB)

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

5. Nesta tela, é possível alterar o nome do pool ZFS, o nome do conjunto de dados, o tamanho do pool 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 raiz é alterado para `zfsBE`.

```

                ZFS Pool Name: rpool
        ZFS Root Dataset Name: zfsBE
        ZFS Pool Size (in MB): 139990
        Size of Swap Area (in MB): 4096
        Size of Dump Area (in MB): 1024
        (Pool size must be between 7006 MB and 139990 MB)
    
```

6. Nesta tela de instalação final, opcionalmente você pode alterar o perfil 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.

=====

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

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

Regions: North America
System Locale: C ( C )

Software: Solaris 10, Entire Distribution
Pool Name: rpool
Boot Environment Name: zfsBE
Pool Size: 139990 MB
Devices in Pool: c1t0d0
                  c1t1d0
```

7. Quando a instalação estiver concluída, verifique as informações resultantes do pool de armazenamento e do sistema de arquivos ZFS. Por exemplo:

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

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

errors: No known data errors
# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
rpool         10.1G  124G   106K   /rpool
rpool/ROOT    5.01G  124G   31K    legacy
rpool/ROOT/zfsBE 5.01G  124G   5.01G   /
rpool/dump    1.00G  124G   1.00G   -
rpool/export  63K    124G   32K    /export
rpool/export/home 31K    124G   31K    /export/home
rpool/swap    4.13G  124G   4.00G   -
```

A saída `zfs list` de amostra identifica os componentes do pool raiz, como o diretório `rpool/ROOT`, que não está acessível por padrão.

8. Para criar outro ambiente de inicialização (BE) do ZFS no mesmo pool de armazenamento, use 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:

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

```
# 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.
```

Criando um BE do ZFS dentro do mesmo pool utiliza recursos de clone e instantâneos do ZFS para criar instantaneamente o BE. Para mais detalhes sobre a utilização do Live Upgrade em uma migração de raiz ZFS, consulte [“Migrando um Sistema de Arquivos de Raiz do ZFS ou Atualizando um Sistema de Arquivos de Raiz do ZFS \(Live Upgrade\)”](#) na página 129.

9. 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                  10.1G  124G   106K   /rpool
rpool/ROOT              5.00G  124G   31K    legacy
rpool/ROOT/zfs2BE       218K   124G   5.00G   /
rpool/ROOT/zfsBE        5.00G  124G   5.00G   /
rpool/ROOT/zfsBE@zfs2BE  104K    -    5.00G   -
rpool/dump              1.00G  124G   1.00G   -
rpool/export            63K    124G   32K    /export
rpool/export/home       31K    124G   31K    /export/home
rpool/swap              4.13G  124G   4.00G   -
```

10. Se desejar inicializar a partir de BE alternativo, utilize o comando `luactivate`.

- SPARC - Use o comando `boot -L` para identificar os ambientes de inicialização disponíveis quando o dispositivo de inicialização contiver um pool de armazenamento do ZFS.

EXEMPLO 4-1 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (Continuação)

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
```

- x86 – Identifique o BE a ser inicializado a partir do menu do GRUB.

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 159](#).

▼ Como migrar um pool raiz espelhado do ZFS (pós-instalação)

Se não for criado um pool raiz espelhado do ZFS durante a instalação, será possível criar um facilmente após a instalação.

Para obter informações sobre a substituição de um disco no pool raiz do ZFS, consulte [“Como substituir um disco no pool raiz ZFS” na página 166](#).

1 Exibe o status atual do pool raiz.

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

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

errors: No known data errors
```

2 Anexe o segundo disco para configurar um pool raiz espelhado.

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


3 Visualize o estado do pool raiz para confirmar que o processo de 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	
clt0d0s0	ONLINE	0	0	0	
clt1d0s0	ONLINE	0	0	0	3.18G resilvered

errors: No known data errors

Na saída anterior, o processo de resilver não está concluído. Resilvering está concluído quando é possível visualizar mensagens semelhantes à seguinte:

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

4 Verifique se é possível inicializar com êxito a partir do segundo disco.

5 Se necessário, configure o sistema para inicializar automaticamente a partir do novo disco.

- SPARC - Use o comando `eeprom` ou o comando `setenv` a partir do PROM de inicialização SPARC para redefinir o dispositivo de inicialização padrão.
- x86 - reconfigure a BIOS do sistema.

Instalando um sistema de arquivos raiz ZFS (instalação de arquivo do Oracle Solaris Flash)

A partir do Solaris versão 10 10/09, é possível criar um arquivo Flash em um sistema com um sistema de arquivos raiz UFS ou ZFS. O arquivo flash de um pool raiz do ZFS contém toda a hierarquia de pools, exceto para volumes de permuta e despejo 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:

- Crie um arquivo flash que possa ser utilizado para instalar e inicializar um sistema com um sistema de arquivos raiz ZFS.
- Efetue uma instalação JumpStart ou uma instalação inicial de um sistema de clone utilizando um arquivo flash ZFS. A criação de um arquivo flash ZFS clona um pool raiz inteiro, e não ambientes de inicialização individuais. É possível excluir os conjuntos de dados individuais do pool usando a opção `-D` para os comandos `flarcreate` e `flar`.

Revise as seguintes limitações antes de considerar a instalação de um sistema com um arquivo Flash ZFS:

- A partir do Oracle Solaris versão 10 8/11, você pode usar a opção de arquivo flash da instalação interativa para instalar um sistema com um sistema de arquivos raiz ZFS. Além disso, você pode usar um arquivo flash para atualizar um BE ZFS alternativo usando o comando `luupgrade`.
 - Um sistema executando a versão Solaris 10 9/10 deve adicionar o patch 124630-51 (SPARC) ou o patch 124631-51 (x86) para instalar um arquivo flash em uma ABE.
 - O sistema mestre, onde você está criando o arquivo flash e o arquivo clone, deve estar no mesmo nível que o patch do kernel. Por exemplo, se você criar um arquivo flash ZFS em um sistema executando a versão Solaris 10 8/11, verifique se o sistema clone também está sendo executado no mesmo nível de patch de kernel que o Solaris 10 8/11. Caso contrário, pode ocorrer falha na instalação do arquivo flash com erros do comando `zfs receive`.
 - Um sistema mestre que executa a versão Solaris 10 9/10 com sistemas de arquivos de raiz descendente, como um sistema de arquivos `/var` separado, deve ser atualizado para o Solaris 10 8/11 antes da criação do arquivo flash e aplicação dele a uma ABE. Caso contrário, ocorrerá falha na instalação do arquivo flash.
 - 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 `sun4v` não pode ser instalado em um sistema `sun4u`.
 - 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 instalar um arquivo UFS/ZFS híbrido.
 - A partir do Solaris versão 10 8/11, você pode usar um arquivo flash UFS para instalar um sistema de arquivos raiz ZFS. Por exemplo:
 - Se você usar a palavra-chave `pool` no perfil do JumpStart, o arquivo flash UFS será instalado em um pool raiz do ZFS.

```
pool rpool auto auto auto mirror c0t0d0s0 c0t1d0s0
```

 - Durante a instalação interativa de um arquivo flash UFS, selecione ZFS como o tipo do sistema de arquivos.
- Embora todo o pool raiz, menos os conjuntos de dados explicitamente excluídos, seja arquivado e instalado, apenas o ambiente de inicialização do ZFS que é inicializado quando o arquivo é criado é utilizável após a instalação do arquivo flash. No entanto, pools que são arquivados com a opção `-A rootdir` do comando `flarcreate` ou `flar` podem ser usados para arquivar um pool raiz diferente do pool raiz que está inicializado atualmente.
- As opções do comando `flarcreate` e `flar` que são usados 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 zfs10upflar
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:

- Crie o arquivo Flash ZFS com o comando `flarcreate` no sistema mestre. Todos os conjuntos de dados no pool 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 seguintes opções de arquivamento são suportadas para a instalação do pool raiz do ZFS com um arquivo flash:

- Utilize o comando `flarcreate` ou `flar` para criar um arquivo flash a partir do pool raiz do ZFS especificado. Se não for especificado, um arquivo Flash do pool raiz padrão é criado.
- Utilize `flarcreate -D dataset` para excluir o conjunto de dados especificado 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, exceto 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 implantados.

EXEMPLO 4-2 Instalação de um sistema com um arquivo Flash do ZFS (instalação JumpStart)

Depois da instalação ou atualização de um sistema mestre com no mínimo o Solaris versão 10 10/09, crie um arquivo flash do pool 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 6.77GB.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.
```

EXEMPLO 4-2 Instalação de um sistema com um arquivo Flash do ZFS (instalação JumpStart)
(Continuação)

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
```

EXEMPLO 4-3 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (instalação de arquivo Flash)

É possível instalar um sistema de arquivos raiz ZFS selecionando uma opção de instalação Flash. Esta opção considera que um arquivo flash ZFS já tenha sido criado e está disponível.

1. Na tela Instalação interativa do Solaris, selecione a opção `F4_Flash`.
2. Na tela Reinicializar após instalação, selecione a opção Reinicialização automática ou Reinicialização manual.
3. Na tela Escolher tipo de sistema de arquivos, selecione ZFS.
4. Na tela Método de recuperação de arquivo Flash, selecione o método de recuperação, como HTTP, FTP, NFS, Arquivo Local, Fita Local ou Dispositivo Local.

Por exemplo, selecione NFS se o arquivo flash for compartilhado de um servidor NFS.

5. Na tela Inclusão de arquivo Flash, especifique o local do arquivo flash ZFS.

Por exemplo, se o local for um servidor NFS, identifique o servidor por seu endereço IP e depois especifique o caminho para o arquivo flash ZFS.

NFS Location: `12.34.567.890:/export/zfs10upflar`

6. Na tela Seleção de arquivo Flash, confirme o método de recuperação e o nome do BE ZFS.

Flash Archive Selection

You selected the following Flash archives to use to install this system. If you want to add another archive to install select "New".

Retrieval Method	Name
NFS	zfsBE

7. Reveja o próximo conjunto de telas, semelhante a uma instalação inicial, e selecione as opções que correspondem à sua configuração:

- Selecionar Discos
- Deseja preservar os dados?

EXEMPLO 4-3 Instalação inicial de um sistema de arquivos raiz ZFS inicializável (instalação de arquivo Flash) (Continuação)

- Definir configurações do ZFS

Reveja as informações de resumo e depois selecione a opção Continuar.

Por exemplo:

Configure ZFS Settings

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

```

ZFS Pool Name: rpool
ZFS Root Dataset Name: s10zfsBE
ZFS Pool Size (in MB): 69995
Size of Swap Area (in MB): 2048
Size of Dump Area (in MB): 1024
(Pool size must be between 7591 MB and 69995 MB)

```

Se o arquivo flash for um fluxo de envio do ZFS, opções do sistema de arquivos combinadas ou separadas /var não estarão presentes. Neste caso, se /var é combinado ou não dependente de como está configurado no sistema mestre.

- Pressione Continuar em Montar sistemas de arquivos remotos ? tela.
- Reveja a tela Perfil e pressione F4 para fazer alterações. Caso contrário, pressione Begin_Installation (F2).

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: Flash
Boot Device: c1t0d0
Root File System Type: ZFS
Client Services: None

Software: 1 Flash Archive
NFS: zfsBE
Pool Name: rpool
Boot Environment Name: s10zfsBE
Pool Size: 69995 MB
Devices in Pool: c1t0d0

```

Instalando um sistema de arquivos raiz ZFS (Instalação com JumpStart)

É possível criar um perfil JumpStart para instalar um sistema de arquivos raiz ZFS ou um sistema de arquivos raiz UFS.

Um perfil JumpStart específico do ZFS deve conter a nova palavra-chave `pool`. A palavra-chave `pool` instala um novo pool raiz e, por padrão, é criado um novo ambiente de inicialização (BE). 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 a utilização dos recursos do JumpStart, consulte [Guia de instalação do Oracle Solaris 10 1/13: instalações JumpStart](#).

Se for configurar zonas depois da instalação inicial JumpStart de um sistema de arquivos raiz ZFS e tiver planos de atualizar ou instalar patches no sistema, consulte “[Usando o Live Upgrade para migrar ou atualizar um sistema com zonas \(Solaris 10 10/08\)](#)” na página 139 ou “[Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas \(no mínimo Solaris 10 5/09\)](#)” na página 144.

Palavras-chave do JumpStart para ZFS

Em um perfil JumpStart 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 pool é alocado, dadas as limitações, como o tamanho dos discos, os segmentos preservados, e assim por diante.

Por exemplo, se você especificar `c0t0d0s0`, o segmento de pool raiz é criado com o maior tamanho possível tanto se especificar a palavra-chave `all` ou `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 pool raiz ZFS porque os pools 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 ]
```

	<code>installbe</code>	Cria e instala um novo BE identificado pela opção <code>bename</code> e pela entrada <i>BE-name</i> .
	<code>bename nome-do-BE</code>	Identifica o <i>nome-do-BE</i> a ser instalado. Se <code>bename</code> não for usado com a palavra-chave <code>pool</code> , é criado um BE padrão.
	<code>dataset mount-point</code>	Use a palavra-chave <code>dataset</code> opcional para identificar o conjunto de dados <code>/var</code> , que é diferente do conjunto de dados raiz. O valor do <i>mount-point</i> está atualmente limitado a <code>/var</code> . Por exemplo, a linha de sintaxe de <code>bootenv</code> de um conjunto de dados <code>/var</code> diferente é semelhante à seguinte: <code>bootenv installbe bename zfsroot dataset /var</code>
<code>pool</code>	Define o novo pool raiz que será criado. Deve ser fornecida a seguinte sintaxe da palavra-chave: <code>pool poolname poolsize swapsize dumpsize vdevlist</code>	
	<i>poolname</i>	Identifica o nome do pool que será criado. O pool é criado com o pool especificado <i>poolsize</i> e com os dispositivos físicos especificados com um ou mais dispositivos <i>vdevlist</i> . O valor <i>poolname</i> não deve identificar o nome de um pool existente porque o pool existente será substituído.
	<i>poolsize</i>	Especifica o tamanho do pool que será criado. O valor pode ser <code>auto</code> ou <code>existing</code> . O valor <code>auto</code> aloca o maior tamanho possível do pool, dadas as limitações, como o tamanho dos discos, e assim por diante. O tamanho adotado é em MB, a menos que esteja especificado por <code>g</code> (GB).
	<i>swapsize</i>	Especifica o tamanho do volume de intercâmbio que será criado. O valor <code>auto</code> significa que o tamanho de permuta padrão é utilizado. É possível especificar um tamanho com um valor <i>size</i> . O tamanho adotado é em MB, a menos que estiver especificado por <code>g</code> (GB).
	<i>dumpsize</i>	Especifica o tamanho do volume de dump que será criado. O valor <code>auto</code> significa que o tamanho de despejo padrão é utilizado. É possível especificar um tamanho com um valor <i>size</i> . O tamanho adotado é em MB, a menos que esteja especificado por <code>g</code> (GB).
	<i>vdevlist</i>	Especifica um ou mais dispositivos que serão usados para criar o pool. O formato de <i>vdevlist</i> é o mesmo que o formato do comando <code>zpool create</code> . Atualmente, somente as configurações espelhadas são suportadas quando vários dispositivos são especificados. Os

dispositivos de *vdevlist* devem ser segmentos para o pool raiz. O valor *any* significa que o software de instalação seleciona um dispositivo adequado.

É possível espelhar quantos discos quiser, mas o tamanho do pool 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 47.](#)

Exemplos de perfil do JumpStart para ZFS

Esta seção oferece exemplos de perfis específicos do ZFS no JumpStart.

O perfil seguinte realiza uma instalação inicial especificada com `install_type` `instalação_inicial` em um novo pool, identificado com `pool` `newpool`, cujo tamanho é definido automaticamente 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 ambiente de inicialização 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 pool 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 pool raiz ZFS. Esta configuração não é recomendável para os sistemas de produção. No entanto, pode ser usada para transição ou necessidades de migração em um sistema pequeno, 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 pool com segmentos de disco e não com o disco inteiro, conforme descrito no [“Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS” na página 111](#). 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 de Raiz do ZFS ou Atualizando um Sistema de Arquivos de Raiz do ZFS (Live Upgrade)

Os recursos relacionados do Live Upgrade estão disponíveis e, se estiverem relacionados aos componentes UFS, funcionam como nas versões anteriores do .

Estão disponíveis os recursos seguintes:

- **Migração do BE UFS para BE ZFS**
 - 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.
 - No Oracle Solaris versão 10 8/11, você pode especificar um sistema de arquivos separado /var ao migrar seu sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS
 - O processo básico para migrar de um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS é o seguinte:

1. Instale os patches necessários do Live Upgrade, se necessário.
2. Instale uma versão atual do Oracle Solaris 10 (Solaris 10 10/08 para Oracle Solaris 10 8/11) ou utilize o programa de atualização padrão para atualizar de uma versão anterior do Oracle Solaris 10 em qualquer sistema suportado com base em SPARC ou x 86.
3. Quando estiver executando o Solaris 10 10/08, crie um pool de armazenamento ZFS para o sistema de arquivos raiz ZFS.
4. Use o Live Upgrade para migrar o sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS.
5. Ative o BE do ZFS com o comando `luactivate`.

■ **Correção ou atualização de um BE ZFS**

- É possível utilizar o comando `luupgrade` para corrigir ou atualizar um BE ZFS. Você também pode usar `luupgrade` para atualizar um BE ZFS alternativo com um arquivo flash ZFS. Para obter informações, consulte o [Exemplo 4–8](#).
- O Live Upgrade pode utilizar os recursos de clone e instantâneo ZFS ao criar um BE do ZFS no mesmo pool. Consequentemente, a criação do ambiente de inicialização é muito mais rápida do que nas versões anteriores.

- **Suporte para migração de zona**– É possível migrar um sistema com zonas, mas as configurações suportadas são limitadas no Solaris versão 10 10/08. Mais configurações de zonas são suportadas a partir da versão do Solaris 10 5/09. Para obter mais informações, consulte as seguintes seções:

- “Usando o Live Upgrade para migrar ou atualizar um sistema com zonas (Solaris 10 10/08)” na página 139
- “Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas (no mínimo Solaris 10 5/09)” na página 144

Se for migrar para um sistema de arquivos raiz ZFS sem zonas, consulte “Usando o Live Upgrade para migrar ou atualizar um sistema de arquivos raiz do ZFS (sem zonas)” na página 132.

Para obter informações detalhadas sobre a instalação do Oracle Solaris e os recursos do Live Upgrade, consulte o *Guia de instalação do Oracle Solaris 10 1/13: Live Upgrade e planejamento da atualização*.

Para informações sobre requisitos do ZFS e do Live Upgrade, consulte “Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS” na página 111.

Problemas de migração ZFS com o Live Upgrade

Revise a lista de problemas a seguir antes de utilizar o 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 um sistema de arquivos raiz ZFS. Para migrar de um sistema de arquivos UFS, você deve usar o Live Upgrade.
- Você deve criar o pool de armazenamento do ZFS que será usado para a inicialização antes da operação com o 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 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 rotulação não altera o esquema de particionamento. Na maioria dos casos, toda capacidade do disco deve estar nos segmentos destinados ao pool 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 ambientes de inicialização do ZFS com o comando `zfs rename`, porque o Live Upgrade não pode detectar a alteração de nome. Os comando subsequentes, como `ludelete`, falharão. De fato, não renomeie os pools ou sistemas de arquivos ZFS se tiver BEs que deseja continuar usando.
- Ao criar um ambiente de inicialização alternativo que seja um clone do ambiente de inicialização primário, você não pode usar as opções `-f`, `-x`, `-y`, `-Y` e `-z` para incluir ou excluir arquivos de tal ambiente de inicialização primário. É possível usar as opções de inclusão e exclusão definidas nos seguintes casos:

```
UFS -> UFS
UFS -> ZFS
ZFS -> ZFS (different pool)
```

- Embora seja possível usar o Live Upgrade para atualizar um sistema de arquivos raiz UFS para um ZFS, não é possível usá-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.
- Se você quiser que o dispositivo de permuta e dump do seu sistema fique em um pool sem raiz, consulte [“Personalizando Volumes de Permuta e Dump do ZFS” na página 157](#).

Usando o Live Upgrade para migrar ou atualizar um sistema de arquivos raiz do ZFS (sem zonas)

Os exemplos a seguir mostram como migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS e como atualizar um sistema de arquivos raiz ZFS.

Se for migrar ou atualizar um sistema com zonas, consulte as seguintes seções:

- [“Usando o Live Upgrade para migrar ou atualizar um sistema com zonas \(Solaris 10 10/08\)” na página 139](#)
- [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas \(no mínimo Solaris 10 5/09\)” na página 144](#)

EXEMPLO 4-4 Usando o Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS

O exemplo seguinte mostra como migrar um sistema de arquivos raiz ZFS 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 para que a operação `lucreate` seja executada.

O pool 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 rotulação não altera o esquema de particionamento. Na maioria dos casos, toda capacidade do disco deve estar nos segmentos destinados ao pool 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.
```

EXEMPLO 4-4 Usando o Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS (Continuação)

```
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:

```
# 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
```

EXEMPLO 4-4 Usando o Live Upgrade para migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS *(Continuação)*

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 Name	Is Complete	Active Now	Active On Reboot	Can Delete	Copy Status
ufsBE	yes	no	no	yes	-
zfsBE	yes	yes	yes	no	-

Se voltar ao BE do UFS, você terá que reimportar os pools 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 4-5 Using Live Upgrade to Create a ZFS BE From a UFS BE (With a Separate /var)

No Oracle Solaris versão 10 8/11, você pode usar a opção `lucreate -D` para identificar que você deseja um sistema de arquivos /var separado criado ao migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS. No exemplo a seguir, o BE UFS existente é migrado para um BE ZFS com um sistema de arquivos /var separado.

```
# lucreate -n zfsBE -p rpool -D /var
Determining types of file systems supported
Validating file system requests
Preparing logical storage devices
Preparing physical storage devices
Configuring physical storage devices
Configuring logical storage devices
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <c0t0d0s0>.
Current boot environment is named <c0t0d0s0>.
Creating initial configuration for primary boot environment <c0t0d0s0>.
INFORMATION: No BEs are configured on this system.
The device </dev/dsk/c0t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <c0t0d0s0> PBE Boot Device </dev/dsk/c0t0d0s0>.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c0tld0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfsBE>.
Source boot environment is <c0t0d0s0>.
Creating file systems on boot environment <zfsBE>.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsBE>.
```

EXEMPLO 4-5 Using Live Upgrade to Create a ZFS BE From a UFS BE (With a Separate /var)
(Continuação)

```

Creating <zfs> file system for </var> in zone <global> on <rpool/ROOT/zfsBE/var>.
Populating file systems on boot environment <zfsBE>.
Analyzing zones.
Mounting ABE <zfsBE>.
Generating file list.
Copying data from PBE <c0t0d0s0> to ABE <zfsBE>
100% of filenames transferred
Finalizing ABE.
Fixing zonepaths in ABE.
Unmounting ABE <zfsBE>.
Fixing properties on ZFS datasets in ABE.
Reverting state of zones in PBE <c0t0d0s0>.
Making boot environment <zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-iaf.mnt
updating /.alt.tmp.b-iaf.mnt/platform/sun4u/boot_archive
Population of boot environment <zfsBE> successful.
Creation of boot environment <zfsBE> successful.
# luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsBE>.
.
.
.
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
# init 6

```

Revise os sistemas de arquivos ZFS recém-criados. Por exemplo:

```

# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               6.29G  26.9G  32.5K  /rpool
rpool/ROOT                          4.76G  26.9G   31K  legacy
rpool/ROOT/zfsBE                    4.76G  26.9G  4.67G  /
rpool/ROOT/zfsBE/var                89.5M  26.9G  89.5M  /var
rpool/dump                          512M   26.9G  512M   -
rpool/swap                          1.03G  28.0G   16K   -

```

EXEMPLO 4-6 Usando o 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 ambiente de inicialização atual estiver no mesmo pool ZFS, a opção -p é omitida.

Se possuir vários BEs de ZFS, faça o seguinte para selecionar a partir de qual BE inicializar:

- SPARC: É possível usar o comando `boot -L` para identificar os BEs disponíveis. Em seguida, selecione um BE do qual inicializar usando o comando `boot -Z`.
- x86: você pode selecionar um BE do menu GRUB.

Para obter mais informações, consulte o [Exemplo 4-12](#).

EXEMPLO 4-6 Usando o Live Upgrade para criar um BE do ZFS a partir de um BE do ZFS
(Continuação)

```
# 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 4-7 Atualização do BE do ZFS (luupgrade)

Você pode atualizar o ambiente de inicialização 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 patches.

```
# lustatus
Boot Environment      Is      Active Active   Can      Copy
Name                 Complete Now    On Reboot Delete Status
-----
zfsBE                 yes     no     no       yes      -
zfs2BE                yes     yes    yes      no       -
# luupgrade -p -n zfsBE -s /net/system/export/s10up/Solaris_10/Product SUNWchxge
Validating the contents of the media </net/install/export/s10up/Solaris_10/Product>.
Mounting the BE <zfsBE>.
Adding packages to the BE <zfsBE>.

Processing package instance <SUNWchxge> from </net/install/export/s10up/Solaris_10/Product>

Chelsio N110 10GE NIC Driver(sparc) 11.10.0,REV=2006.02.15.20.41
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.

This appears to be an attempt to install the same architecture and
```


EXEMPLO 4-7 Atualização do BE do ZFS (luupgrade) (Continuação)

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.
```

Ou você pode criar um novo BE para atualizar para uma versão mais recente do Oracle Solaris. Por exemplo:

```
# luupgrade -u -n newBE -s /net/install/export/s10up/latest
```

Onde a opção -s especifica a localização de um meio de instalação do Solaris.

EXEMPLO 4-8 Criação de um BE ZFS com um arquivo Flash ZFS (luupgrade)

No Oracle Solaris versão 10 8/11, você pode usar o comando `luupgrade` para criar um BE ZFS a partir de um arquivo flash ZFS existente. O processo básico realiza-se da seguinte forma:

1. Crie um arquivo flash xde um sistema mestre com um BE ZFS.

Por exemplo:

```
master-system# flarcreate -n s10zfsBE /tank/data/s10zfsflar
Full Flash
Checking integrity...
Integrity OK.
Running precreation scripts...
Precreation scripts done.
Determining the size of the archive...
The archive will be approximately 4.67GB.
Creating the archive...
Archive creation complete.
Running postcreation scripts...
Postcreation scripts done.
```

EXEMPLO 4-8 Criação de um BE ZFS com um arquivo Flash ZFS (luupgrade) (Continuação)

```
Running pre-exit scripts...
Pre-exit scripts done.
```

2. Disponibilize o arquivo flash ZFS que foi criado no sistema mestre para o sistema de clone.

Locais possíveis de arquivo são um sistema de arquivos local, HTTP, FTP, NFS, e assim por diante.

3. Crie um BE ZFS alternativo vazio no sistema de clone.

Use a opção `-s` para especificar que este é um BE vazio a ser preenchido com o conteúdo do arquivo flash ZFS.

Por exemplo:

```
clone-system# lucreate -n zfsflashBE -s - -p rpool
Determining types of file systems supported
Validating file system requests
Preparing logical storage devices
Preparing physical storage devices
Configuring physical storage devices
Configuring logical storage devices
Analyzing system configuration.
No name for current boot environment.
INFORMATION: The current boot environment is not named - assigning name <s10zfsBE>.
Current boot environment is named <s10zfsBE>.
Creating initial configuration for primary boot environment <s10zfsBE>.
INFORMATION: No BEs are configured on this system.
The device </dev/dsk/c0t0d0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <s10zfsBE> PBE Boot Device </dev/dsk/c0t0d0s0>.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/c0t1d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating <zfs> file system for </> in zone <global> on <rpool/ROOT/zfsflashBE>.
Creation of boot environment <zfsflashBE> successful.
```

4. Instale o arquivo flash ZFS no BE alternativo.

Por exemplo:

```
clone-system# luupgrade -f -s /net/server/export/s10/latest -n zfsflashBE -a /tank/data/zfs10up2flar
miniroot filesystem is <lofs>
Mounting miniroot at </net/server/s10up/latest/Solaris_10/Tools/Boot>
Validating the contents of the media </net/server/export/s10up/latest>.
The media is a standard Solaris media.
Validating the contents of the miniroot </net/server/export/s10up/latest/Solaris_10/Tools/Boot>.
Locating the flash install program.
Checking for existence of previously scheduled Live Upgrade requests.
Constructing flash profile to use.
Creating flash profile for BE <zfsflashBE>.
Performing the operating system flash install of the BE <zfsflashBE>.
CAUTION: Interrupting this process may leave the boot environment unstable or unbootable.
Extracting Flash Archive: 100% completed (of 5020.86 megabytes)
The operating system flash install completed.
updating /.alt.tmp.b-rgb.mnt/platform/sun4u/boot_archive

The Live Flash Install of the boot environment <zfsflashBE> is complete.
```

EXEMPLO 4-8 Criação de um BE ZFS com um arquivo Flash ZFS (luupgrade) (Continuação)

5. Ative o ambiente de inicialização alternativo.

```
clone-system# luactivate zfsflashBE
A Live Upgrade Sync operation will be performed on startup of boot environment <zfsflashBE>.
.
.
.
Modifying boot archive service
Activation of boot environment <zfsflashBE> successful.
```

6. Reinicialize o sistema.

```
clone-system# init 6
```

Usando o Live Upgrade para migrar ou atualizar um sistema com zonas (Solaris 10 10/08)

É possível utilizar o Live Upgrade para migrar um sistema com zonas, 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 zonas são suportadas. Para obter mais informações, consulte [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas \(no mínimo Solaris 10 5/09\)”](#) na página 144.

Esta seção descreve como instalar e configurar um sistema com zonas para que possa ser atualizado e ter patches instalados com o Live Upgrade. Se for migrar para um sistema de arquivos raiz ZFS sem zonas, consulte [“Usando o Live Upgrade para migrar ou atualizar um sistema de arquivos raiz do ZFS \(sem zonas\)”](#) na página 132.

Se for migrar ou configurar um sistema com zonas no Solaris 10 10/08, consulte os seguintes procedimentos:

- [“Como migrar um sistema de arquivos raiz UFS com raízes de zona no UFS para um sistema de arquivos raiz ZFS \(Solaris 10 10/08\)”](#) na página 140
- [“Como configurar um sistema de arquivos raiz ZFS com raízes de zona em ZFS \(Solaris 10 10/08\)”](#) na página 141
- [“Como fazer upgrade ou aplicar patches em um sistema de arquivos raiz ZFS com raízes de zonas em ZFS \(Solaris 10 10/08\)”](#) na página 143
- [“Resolvendo problemas do ponto de montagem ZFS que impedem a inicialização com êxito \(Solaris 10 10/08\)”](#) na página 163

Siga esses procedimentos recomendados para configurar zonas em um sistema com sistemas de arquivos raiz ZFS para garantir que o Live Upgrade possa ser usado em tal sistema.

▼ Como migrar um sistema de arquivos raiz UFS com raízes de zona no 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 zonas instaladas para um sistema de arquivos de raiz e configuração de raiz de zona ZFS que pode ser atualizada ou ter patches instalados.

Nas etapas a seguir, o nome do pool de exemplo é `rpool` e os nomes de exemplo do ambiente de inicialização (BEs) ativo começa com `s10BE*`.

1 Atualize os sistema para o Solaris 10 10/08 se estiver executando uma versão anterior do Solaris 10.

Para obter mais informações sobre a atualização de um sistema que executa o Solaris versão 10, consulte o [Guia de instalação do Oracle Solaris 10 1/13: Live Upgrade e planejamento da atualização](#).

2 Crie o pool raiz.

```
# zpool create rpool mirror c0t1d0 c1t1d0
```

Para obter informações sobre os requisitos do pool de raiz, consulte “Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS” na página 111.

3 Confirme se as zonas do ambiente do UFS foram inicializadas.

4 Crie o novo ambiente de inicialização do ZFS.

```
# lucreate -n s10BE2 -p rpool
```

Este comando estabelece os conjuntos de dados no pool raiz do novo ambiente de inicialização e copia o ambiente de inicialização atual (incluindo as zonas) 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 zona no UFS ainda estão no sistema de arquivos raiz UFS. É necessário completar as próximas etapas para migrar totalmente as zonas UFS para uma configuração ZFS suportada.

6 Reinicialize o sistema.

```
# init 6
```

7 Migre as zonas para um BE do ZFS.

a. Inicialize as zonas.

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 zonas estão inicializados.

8 Resolva qualquer potencial problema de ponto de montagem.

Devido a um erro no Live Upgrade, o ambiente de inicialização inativo poderá falhar ao ser inicializado porque o conjunto de dados ZFS ou o conjunto de dados ZFS da zona 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/s10up
```

NAME	MOUNTPOINT
rpool/ROOT/s10up	/.alt.tmp.b-VP.mnt/
rpool/ROOT/s10up/zones	/.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10up/zones/zonerootA	/.alt.tmp.b-VP.mnt/zones/zonerootA

O ponto de montagem do ambiente de inicialização ZFS raiz (rpool/ROOT/s10up) 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/s10up
# zfs set mountpoint=/ rpool/ROOT/s10up
```

c. Reinicialize o sistema.

Quando a opção para inicializar um BE específico for apresentada no prompt do OpenBoot PROM ou no menu GRUB, selecione o BE cujos pontos de montagem acabam de ser corrigidos.

▼ Como configurar um sistema de arquivos raiz ZFS com raízes de zona em ZFS (Solaris 10 10/08)

Este procedimento explica como configurar um sistema de arquivos raiz ZFS e a configuração da raiz de zona ZFS que pode ser atualizada ou ter patches instalados. Nesta configuração, as raízes de zona ZFS são criadas como conjuntos de dados ZFS.

Nas etapas a seguir, o nome do pool de exemplo é `rpool` e o nome de exemplo do ambiente de inicialização ativo é `s10BE*`. O nome do conjunto de dados das zonas pode ser qualquer nome de conjunto de dados válido. No exemplo a seguir, o nome do conjunto de dados de zonas é `zones`.

1 Instale o sistema com uma raiz ZFS utilizando o instalador de texto interativo ou o método de instalação JumpStart.

Dependendo do método de instalação escolhido, consulte [“Instalação de um sistema de arquivos raiz ZFS \(instalação inicial do Oracle Solaris\)”](#) na página 114 ou [“Instalando um sistema de arquivos raiz ZFS \(Instalação com JumpStart\)”](#) na página 126.

2 Inicialize o sistema a partir do pool raiz recém-criado.

3 Crie um conjunto de dados para agrupar as raízes da zona.

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 Live Upgrade e do código de inicialização do sistema.

4 Monte o conjunto de dados de zonas 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 zona.

```
# 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 zona.

```
# chmod 700 /zones/zonerootA
```

7 Configure a zona, definindo o caminho da zona 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 zonas para que se inicializem automaticamente quando o sistema for inicializado usando a seguinte sintaxe:

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

8 Instale a zona.

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

9 Inicialize a zona.

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

▼ Como fazer upgrade ou aplicar patches em um sistema de arquivos raiz ZFS com raízes de zonas 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 zonas em ZFS. Estas atualizações podem consistir em uma melhora do sistema quanto na 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 BE do para ser atualizado ou aplicar as correções.

```
# lucreate -n newBE
```

O BE existente, incluindo todas as zonas, é clonado. É criado um conjunto de dados para cada conjunto de dados no 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 correções ao novo ambiente de inicialização:

- Atualize o sistema.

```
# luupgrade -u -n newBE -s /net/install/export/s10up/latest
```

onde a opção -s especifica a localização de um meio de instalação do Oracle Solaris.

- Aplique patches ao 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 BE recém-ativado.

```
# init 6
```

5 Resolva qualquer potencial problema de ponto de montagem.

Devido a um erro no 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 zona 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/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 no prompt do OpenBoot PROM ou no menu GRUB, 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 zonas (no mínimo Solaris 10 5/09)

É possível utilizar o recurso Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas a partir da versão Solaris 10 10/08. Configurações adicionais (esparsa ou completa) são suportadas pelo Live Upgrade a partir do Solaris versão 10 5/09.

Esta seção descreve como configurar e instalar um sistema com zonas para que possa ser atualizado e ter o patch instalado com o Live Upgrade a partir da versão Solaris 10 5/09. Se for migrar para um sistema de arquivos raiz ZFS sem zonas, consulte [“Usando o Live Upgrade para migrar ou atualizar um sistema de arquivos raiz do ZFS \(sem zonas\)”](#) na página 132.

Considere os seguintes pontos ao utilizar o Live Upgrade com ZFS e zonas começando com a versão Solaris 10 5/09:

- Para utilizar o Live Upgrade com configurações de zonas que são suportadas, começando com o Solaris versão 10 5/09, é necessário primeiro atualizar o sistema para o Solaris versão 10 5/09 pelo menos, ao utilizar o programa de upgrade padrão.
- Logo, com o Live Upgrade, é possível migrar o sistema de arquivos raiz UFS com raízes de zona para o sistema de arquivos raiz ZFS ou atualizar ou instalar patches no sistema de arquivos raiz ZFS e raízes de zona.
- Não é possível migrar diretamente configurações de zonas não suportadas de uma versão Solaris 10 anterior à versão Solaris 10 5/09, no mínimo.

Se estiver migrando ou configurando um sistema com zonas começando com a versão Solaris 10 5/09, revise as seguintes informações:

- “ZFS suportado com informações de configuração de raiz de zona (pelo menos Solaris 10 5/09)” na página 145
- “Como criar um BE do ZFS com um sistema de arquivos raiz ZFS e uma raiz de zona (pelo menos Solaris 10 5/09)” na página 147
- “Como atualizar ou instalar patches em um sistema de arquivos raiz ZFS com raízes de zona (pelo menos Solaris 10 5/09)” na página 148
- “Como migrar um sistema de arquivos raiz UFS com raiz de zona para um sistema de arquivos raiz ZFS (pelo menos Solaris 10 5/09)” na página 152

ZFS suportado com informações de configuração de raiz de zona (pelo menos Solaris 10 5/09)

Revise as configurações de zona suportadas antes de utilizar o recurso Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas.

- **Migrar um sistema de arquivos raiz UFS para um sistema de arquivos raiz ZFS** – São suportadas as seguintes configurações de raízes de zona:
 - 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 zona 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 zona

Um sistema de arquivos raiz UFS que possui uma raiz de zona como um ponto de montagem não é suportado.

- **Migrar ou atualizar um sistema de arquivos raiz ZFS** – São suportadas as seguintes configurações de raízes de zona:
 - Em um sistema de arquivos em um pool raiz ZFS ou não-raiz. Por exemplo, /zonepool/zones é aceitável. Em alguns casos, se um sistema de arquivos para a raiz da zona não for fornecido antes da operação do Live Upgrade, um sistema de arquivos para a raiz de zona (zoneds) será criado pelo Live Upgrade.

- Em um sistema de arquivos descendente ou em um subdiretório de um sistema de arquivos ZFS, desde que os caminhos de área não estejam aninhados. Por exemplo, /zonepool/zones/zone1 e /zonepool/zones/zone1_dir são aceitáveis.

No exemplo a seguir, zonepool/zones é um sistema de arquivos que contém as raízes de zona e rpool contém o BE do ZFS:

```
zonepool
zonepool/zones
zonepool/zones/myzone
rpool
rpool/ROOT
rpool/ROOT/myBE
```

O Live Upgrade faz instantâneos e clones das zonas 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 anterior, se /zonepool/zones fosse um subdiretório e não um sistema de arquivos separado, então o Live Upgrade iria migrá-lo como um componente do pool raiz, rpool.

- **A seguinte configuração de ZFS e de zona não é suportada:**
 - O Live upgrade não pode ser usado para criar um BE alternativo quando o BE de origem tem uma zona não global com um caminho de zona definido para o ponto de montagem de um sistema de arquivos do pool de nível superior. Por exemplo, se o pool zonepool tiver um sistema de arquivos montado como /zonepool, você não poderá ter uma área não global com um caminho de zona definido como /zonepool.
 - Não adicione uma entrada do sistema de arquivos para uma zona não global no arquivo /etc/vfstab da zona global. Em vez disso, use o recurso add fs de zonecfg para adicionar um sistema de arquivos a uma zona não global.
- **Migração de zonas ou informações de atualização com zonas para UFS e ZFS:** observe os seguintes fatores que podem afetar tanto a migração quanto a atualização de ambientes UFS e ZFS:
 - Se tiver configurado as zonas conforme descrito em [“Usando o Live Upgrade para migrar ou atualizar um sistema com zonas \(Solaris 10 10/08\)”](#) na página 139 na versão do Solaris 10 10/08 e atualizado para pelo menos o Solaris 10 5/09, você poderá 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 zona 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 zona (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 comando `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 zona 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 zonas 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>.
.
.
.
```

5 Ative o ambiente de inicialização ZFS.

```
# init 6
```

6 Comprove se os sistemas de arquivos ZFS e as zonas 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 patches em um sistema de arquivos raiz ZFS com raízes de zona (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 zona na versão Solaris 10 5/09, no mínimo. Estas atualizações podem ser tanto uma melhora do sistema quanto a aplicação de patches.

Nas etapas a seguir, zfs2BE é o nome de exemplo do ambiente de inicialização que é atualizado ou tem patches instalados.

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 zonas 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 aplicar correções 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 Oracle Solaris.

Este processo pode demorar um longo tempo.

Para obter um exemplo completo do processo de luupgrade, consulte o [Exemplo 4-9](#).

- Aplique patches ao 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 4–9 Atualização de um sistema de arquivos raiz ZFS com raiz de zona para um sistema de arquivos raiz ZFS do Oracle Solaris 10 9/10

Neste exemplo, um ambiente de inicialização ZFS (zfsBE), criado em um sistema Solaris versão 10 10/09 com um sistema de arquivos raiz ZFS e uma raiz de zona em um pool não-raiz, é atualizado para o Oracle Solaris versão 10 9/10. Este processo pode ser demorado. A seguir, o BE atualizado (zfs2BE) é ativado. Certifique-se de que as zonas estejam instaladas e inicializadas antes de tentar atualizar.

Neste exemplo, o grupo zonepool e o conjunto de dados /zonepool/zones e a zona 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 zfs2BE
# init 6
# lustatus
Boot Environment          Is      Active Active   Can    Copy
Name                      Complete Now    On Reboot Delete Status
-----
zfsBE                     yes     no     no      yes    -
zfs2BE                    yes     yes    yes     no     -
# zoneadm list -cv
ID NAME          STATUS    PATH                                BRAND  IP
0  global         running   /                                  native shared
-  zfszone        installed /zonepool/zones                  native shared
```

▼ **Como migrar um sistema de arquivos raiz UFS com raiz de zona 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 zona para a versão Solaris 10 5/09, pelo menos. Em seguida, use o Live Upgrade para criar um BE do ZFS.

Nas etapas a seguir, o nome do BE do UFS de exemplo é `clt1d0s0`, a raiz de zona 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 obter informações sobre a atualização de um sistema que executa o Solaris versão 10, consulte o [Guia de instalação do Oracle Solaris 10 1/13: Live Upgrade e planejamento da atualização](#).

2 Crie o pool raiz.

Para obter informações sobre os requisitos do pool raiz, consulte “Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS” na página 111.

3 Verifique se as zonas do ambiente UFS foram 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 ZFS.

```
# lucreate -c clt1d0s0 -n zfsBE -p rpool
```

Este comando estabelece os conjuntos de dados no pool raiz do novo ambiente de inicialização e copia o ambiente de inicialização atual (incluindo as zonas) em tais conjuntos de dados.

5 Ative o novo ambiente de inicialização ZFS.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now    On Reboot Delete Status
-----
clt1d0s0              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 Confirme se os sistemas de arquivos ZFS e as zonas 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-cltld0s0	634M	7.62G	633M	/zonepool/zones-cltld0s0
zonepool/zones-cltld0s0@zfsBE	262K	-	633M	-

```
# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
-	zfszone	installed	/zonepool/zones	native	shared

Exemplo 4–10 Migrando um sistema de arquivos raiz UFS com raiz de zona para um sistema de arquivos raiz ZFS

Neste exemplo, um sistema Oracle Solaris 10 9/10 com um sistema de arquivos raiz UFS e uma raiz de zona (/uzone/ufszone) e um pool não-raiz ZFS (grupo) e a raiz de zona (/grupo/zfszone) é migrado para um sistema de arquivos raiz ZFS. Certifique-se de que o pool raiz ZFS seja criado e que as zonas 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/ufszone	native	shared
3	zfszone	running	/pool/zones/zfszone	native	shared

```
# lucreate -c ufsBE -n zfsBE -p rpool
```

Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <zfsBE>.
Creating initial configuration for primary boot environment <zfsBE>.
The device </dev/dsk/cltld0s0> is not a root device for any boot environment; cannot get BE ID.
PBE configuration successful: PBE name <ufsBE> PBE Boot Device </dev/dsk/cltld0s0>.
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/cltld0s0> 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

Gerenciamento dos dispositivos de permuta e despejo do ZFS

Durante uma instalação inicial do SO Oracle Solaris ou depois de executar uma migração do Live Upgrade de um sistema de arquivos UFS, é criada uma área de permuta em um volume ZFS no pool raiz do 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 SO Oracle Solaris de um sistema de arquivos UFS, é criado um dispositivo de despejo em um volume ZFS no pool 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 você desativar e remover o dispositivo de despejo, deverá ativá-lo com o comando `dumpadm` após sua recriação. 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 dump que são criados pelos programas de instalação, consulte o [“Requisitos de instalação do e Oracle Solaris Live Upgrade para suporte ZFS”](#) na página 111.

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 156.

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 permuta ou o dispositivo de despejo 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 16, “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 156
- [“Solução de problemas para dispositivo de despejo ZFS”](#) na página 158

Ajustando os tamanhos dos dispositivos de permuta e despejo ZFS

Talvez você precise ajustar o tamanho dos dispositivos de permuta e despejo após a instalação ou possivelmente recriar os volumes de permuta e despejo.

- Você pode ajustar o tamanho dos volumes de intercâmbio e dump durante uma instalação inicial. Para obter mais informações, consulte o [Exemplo 4–1](#).
- Você pode criar e dimensionar os volumes de intercâmbio e dump antes de realizar a operação com o 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. Crie seu volume de permuta.

```
# zfs create -V 2G rpool/swap
```

5. É preciso ativar a área de permuta quando um novo dispositivo de permuta é adicionado ou alterado.

```
# swap -a /dev/zvol/dsk/rpool/swap
```

6. Adicione uma entrada para o volume de permuta para o arquivo `/etc/vfstab`.

O Live Upgrade não redimensiona os volumes de intercâmbio e dump 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      -
```

- Se a área de permuta atual não estiver em uso, você poderá redimensionar o tamanho do volume de permuta atual, mas deverá reinicializar o sistema para ver o tamanho de espaço de permuta aumentado.

```
# zfs get volsize rpool/swap
NAME          PROPERTY  VALUE   SOURCE
rpool/swap    volsize   4G      local
```

```
# zfs set volsize=8g rpool/swap
# zfs get volsize rpool/swap
NAME          PROPERTY  VALUE   SOURCE
rpool/swap    volsize   8G      local
# init 6
```

- Você pode tentar redimensionar o volume de permuta, mas talvez seja melhor remover o dispositivo de permuta. Em seguida, recrie-o. Por exemplo:

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs create -V 2g rpool/swap
# swap -a /dev/zvol/dsk/rpool/swap
```

- 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 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.

Personalizando Volumes de Permuta e Dump do ZFS

Tenha o seguinte em mente se você remover os volumes de permuta e dump padrão e recriá-los em um pool que não é de raiz (dados):

- Se você quiser criar dispositivos de permuta e dump em um pool que não é de raiz, não crie volumes de permuta e dump em um pool RAIDZ. Se um pool incluir volumes de permuta e dump, ele deverá ser um pool de um disco ou espelhado.
- Se você usar o Live Upgrade para atualizar seu sistema, use a opção -P para preservar o dispositivo de dump de PBE para ABE. Por exemplo:

```
# lucreate -n newBE -P
```

Solução de problemas para dispositivo de despejo ZFS

Revise o seguinte se tiver problemas ao capturar um despejo de memória de sistema ou ao 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 de despejo 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 obter informações sobre o dimensionamento dos dispositivos de permuta e dump, 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 pool raiz, que não pode possuir vários dispositivos de primeiro nível.

Inicializando 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 pool 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 `bootfs` do pool. Esta seleção padrão pode ser ignorada ao especificar um conjunto de dados inicializável alternativo com o comando `boot - Z`.

Inicializando 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:

- [“Instalação de um sistema de arquivos raiz ZFS \(instalação inicial do Oracle Solaris\)” na página 114](#)
- [“Como migrar um pool raiz espelhado do ZFS \(pós-instalação\)” na página 120](#)

Consulte os problemas conhecidos em relação aos pools raiz ZFS espelhados:

- Se você substituir um disco do pool raiz usando o comando `zpool replace`, deverá instalar as informações de inicialização no disco recém-substituído usando o comando `installboot` ou `installgrub`. Se você criar um pool raiz ZFS espelhado com o método de instalação inicial ou se você usar o comando `zpool attach` para conectar um disco ao pool raiz, então essa etapa será desnecessária. A sintaxe do comando `installboot` e `installgrub` a seguir:
 - SPARC:


```
sparc# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk
```
 - x86:

```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/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 seguinte pool:

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

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

- SPARC: especifique o disco alternativo no prompt ok. Por exemplo:

```
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 pool 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 SO Oracle Solaris e do Live Upgrade, o sistema de arquivos raiz ZFS padrão 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

`bootfs`, 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 baseado em SPARC, há duas opções de inicialização do ZFS 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 pool 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 4-11 SPARC: inicializando a partir de um ambiente de inicialização específico ZFS

Se possuir vários BEs do ZFS em um pool de armazenamento ZFS no dispositivo de inicialização do sistema, é possível utilizar o comando `luactivate` para especificar um BE padrão.

Por exemplo, a seguintes saída `lustatus` mostra que dois BEs ZFS estão disponíveis:

# 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:

```
ok boot -L
Rebooting with command: boot -L
```

EXEMPLO 4-11 SPARC: inicializando a partir de um ambiente de inicialização específico ZFS
(*Continuação*)

```
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 4-12 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 a instalação do SO Oracle Solaris ou durante o processo do Live Upgrade para inicializar o ZFS automaticamente:

```
title Solaris 10 1/13 X86
findroot (rootfs0,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

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 4-13 x86: inicializando um sistema de arquivos ZFS

Quando um sistema é inicializado de um sistema de arquivos ZFS, o dispositivo raiz é especificado pelo parâmetro de inicialização `-B $ZFS-BOOTFS`. Por exemplo:

```
title Solaris 10 1/13 X86
findroot (pool_rpool,0,a)
kernel /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttya
module /boot/x86.miniroot-safe
```

EXEMPLO 4-14 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 (pool_rpool,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 (BE) ativo é usar o comando `luactivate`. Se a inicialização do BE ativo falha devido à uma instalação de patch incorreta ou a um erro de configuração, a única forma de inicializar a partir de outro BE é selecioná-lo no momento da inicialização. Você pode selecionar um BE alternativo inicializando-o explicitamente do PROM em um sistema baseado em SPARC ou do menu do GRUB em um sistema baseado no x86.

Devido a um erro no Live Upgrade no Solaris versão 10 10/08, o ambiente de inicialização inativo poderá falhar ao inicializar porque um conjunto de dados ZFS ou o conjunto de dados ZFS de uma zona no ambiente de inicialização possui 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 ZFS de uma zona possuir um ponto de montagem inválido, tal ponto de montagem poderá 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/s10up
```

NAME	MOUNTPOINT
rpool/ROOT/s10up	/.alt.tmp.b-VP.mnt/
rpool/ROOT/s10up/zones	/.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10up/zones/zonerootA	/.alt.tmp.b-VP.mnt/zones/zonerootA

O ponto de montagem do ambiente de inicialização raiz (rpool/ROOT/s10up) 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/s10up
# zfs set mountpoint=/ rpool/ROOT/s10up
```

5 Reinicialize o sistema.

Quando a opção de inicializar um BE específico for apresentada no prompt do OpenBoot PROM ou no menu GRUB, selecione o ambiente de inicialização cujos pontos de montagem acabam de ser corrigidos.

Inicialização para fins 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.

Você deverá 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 165](#)
- [“Como inicializar o ZFS de uma mídia alternativa” na página 165](#)

Se desejar recuperar a partir de um pool raiz ou de um instantâneo de pool raiz, consulte [“Recuperando o pool raiz ZFS ou instantâneos do pool raiz”](#) na página 166.

▼ Como inicializar o ZFS no modo à prova de falhas

1 Inicialize o modo à prova de falhas.

- Em um sistema baseado em SPARC, digite o seguinte no prompt ok:
`ok boot -F failsafe`
- Em um sistema x86, selecione o modo à prova de falhas no menu 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 ocorrer, você deverá inicializar de um servidor de instalação de rede ou do DVD de instalação do Oracle Solaris. Importe o pool raiz, monte o BE do ZFS e tente resolver o problema.

1 Inicialize de um DVD de instalação ou da rede.

- SPARC - selecione um dos seguintes métodos de inicialização:
`ok boot cdrom -s`
`ok boot net -s`

Se não utilizar a opção -s, deverá sair do programa de instalação.

- x86 – selecione a opção de inicialização de rede ou de inicialização do DVD local.

2 Importe o pool 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 166](#)
- [“Como criar instantâneos do pool raiz” na página 169](#)
- [“How to Re-create a ZFS Root Pool and Restore Root Pool Snapshots” na página 170](#)
- [“Como reverter os instantâneos do pool raiz a partir da inicialização à prova de falhas” na página 172](#)

▼ Como substituir um disco no pool raiz ZFS

Pode ser necessário substituir um disco no pool raiz pelas seguintes razões:

- O pool raiz é muito pequeno e deseja substituir o disco menor por um disco maior.
- O disco do pool raiz apresenta falhas. Em um pool 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 DVD ou a rede, antes de substituir o disco do pool raiz.

Em uma configuração de pool raiz espelhado, é possível tentar substituir um disco sem inicializar de uma mídia alternativa. É possível substituir um disco danificado 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 ver um exemplo de como anexar um disco adicional e desanexar um disco de pool raiz.

Alguns softwares necessitam que traga um disco off-line e desconfigure-o antes de tentar a operação `zpool replace` para substituir um disco danificado. 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
```

Com alguns hardwares, não é necessário colocar o disco on-line ou reconfigurá-lo depois de 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 de procedimento, o nome do caminho para o atual disco de pool 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 as seguintes referências:

- SPARC: “How to Set Up a Disk for a ZFS Root File System” no *System Administration Guide: Devices and File Systems*
- x86 “How to Set Up a Disk for a ZFS Root File System” no *System Administration Guide: Devices and File Systems*

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
clt10d0s0	ONLINE	0	0	0
clt9d0s0	ONLINE	0	0	0

errors: No known data errors

5 Se você estiver substituindo um disco de pool raiz menor por um maior, defina a propriedade autoexpand para expandir o tamanho do pool.

Determine o tamanho do pool rpool existente:

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

```
# zpool set autoexpand=on rpool
```

Examine o tamanho do pool rpool expandido:

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

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 clt10d0s0
```

8 Configure o sistema para inicializar automaticamente a partir do novo disco redefinindo o dispositivo de inicialização padrão.

- SPARC - Use o comando eeprom ou o comando setenv a partir do PROM de inicialização SPARC.

- x86 - Reconfigure a BIOS do sistema.

▼ Como criar instantâneos do pool raiz

É possível criar instantâneos do pool raiz para fins de recuperação. A melhor forma de criar instantâneos de pool raiz é efetuar um instantâneo recursivo do pool raiz.

O procedimento a seguir cria um instantâneo de pool raiz recursivo e armazena o instantâneo como um arquivo e como instantâneos em um pool em um sistema remoto. No caso de uma falha no pool raiz, o conjunto de dados remoto não pode ser montado utilizando o NFS e o arquivo de instantâneo pode ser recebido no pool recriado. Você também pode armazenar instantâneos de pool como os instantâneos reais em um pool 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 Oracle Solaris.

Validando remotamente instantâneos armazenados como arquivos ou instantâneos como um importante passo em uma recuperação de pool raiz. Com ambos os métodos, os instantâneos seriam recriados em uma base de rotina, como quando a configuração do pool é alterada ou quando o SO Solaris 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@snap1
local# zfs list -r rpool
```

#	NAME	USED	AVAIL	REFER	MOUNTPPOINT
	rpool	15.1G	119G	106K	/rpool
	rpool@snap1	0	-	106K	-
	rpool/ROOT	5.00G	119G	31K	legacy
	rpool/ROOT@snap1	0	-	31K	-
	rpool/ROOT/zfsBE	5.00G	119G	5.00G	/
	rpool/ROOT/zfsBE@snap1	0	-	5.00G	-
	rpool/dump	2.00G	120G	1.00G	-
	rpool/dump@snap1	0	-	1.00G	-
	rpool/export	63K	119G	32K	/export

```

rpool/export@snap1      0      -    32K  -
rpool/export/home      31K    119G   31K  /export/home
rpool/export/home@snap1 0      -    31K  -
rpool/swap             8.13G  123G  4.00G -
rpool/swap@snap1       0      -    4.00G -

```

4 Envie os instantâneos do pool raiz ao sistema remoto.

Por exemplo, para enviar os instantâneos do pool raiz para um pool remoto como um arquivo, use sintaxe semelhante à seguinte:

```

local# zfs send -Rv rpool@snap1 > /net/remote-system/rpool/snaps/rpool.snap1
sending from @ to rpool@snap1
sending from @ to rpool/ROOT@snap1
sending from @ to rpool/ROOT/s10zfsBE@snap1
sending from @ to rpool/dump@snap1
sending from @ to rpool/export@snap1
sending from @ to rpool/export/home@snap1
sending from @ to rpool/swap@snap1

```

```

local# zfs send -Rv rpool@snap1 > /net/remote-system/rpool/snaps/rpool.snap1
sending from @ to rpool@snap1
sending from @ to rpool/export@snap1
sending from @ to rpool/export/home@snap1
sending from @ to rpool/ROOT@snap1
sending from @ to rpool/ROOT/zfsBE@snap1
sending from @ to rpool/dump@snap1
sending from @ to rpool/swap@snap1

```

Para enviar os instantâneos do pool raiz para um pool remoto como instantâneos, use sintaxe semelhante à seguinte:

```

local# zfs send -Rv rpool@snap1 | ssh remote-system zfs receive
-Fd -o canmount=off tank/snaps
sending from @ to rpool@snap1
sending from @ to rpool/export@snap1
sending from @ to rpool/export/home@snap1
sending from @ to rpool/ROOT@snap1
sending from @ to rpool/ROOT/zfsBE@snap1
sending from @ to rpool/dump@snap1
sending from @ to rpool/swap@snap1

```

▼ How to Re-create a ZFS Root Pool and Restore Root Pool Snapshots

Neste procedimento, suponha as condições a seguir:

- O pool raiz ZFS não pode ser recuperado.
- Os instantâneos do pool raiz ZFS estão armazenados em um sistema remoto e são compartilhados no NFS.

Todas as etapas são efetuadas no sistema local.

1 Inicialize de um DVD de instalação 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 sistema de arquivos do instantâneo remoto se você tiver enviado os instantâneos do pool raiz como um arquivo para o sistema remoto.

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 pool raiz for substituído e não contiver um rótulo de disco que possa ser utilizada pelo ZFS, será necessário rotular o disco novamente.

Para obter mais informações sobre rotulagem do disco, consulte estas referências:

- SPARC: “How to Set Up a Disk for a ZFS Root File System” no *System Administration Guide: Devices and File Systems*
- x86 “How to Set Up a Disk for a ZFS Root File System” no *System Administration Guide: Devices and File Systems*

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.snap1 | zfs receive -Fdu rpool
```

O uso da opção -u significa que o arquivo restaurado não é montado quando a operação do zfs receive terminar.

Para restaurar os instantâneos do pool raiz real que são armazenados em um pool em um sistema remoto, use sintaxe semelhante à seguinte:

```
# ssh remote-system zfs send -Rb tank/snaps/rpool@snap1 | zfs receive -F rpool
```

6 Verifique se os conjuntos de dados do pool raiz foram restaurados.

Por exemplo:

```
# zfs list
```

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@snap1
# zfs list -r rpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	7.84G	59.1G	109K	/rpool
rpool@snap1	21K	-	106K	-
rpool/ROOT	4.78G	59.1G	31K	legacy
rpool/ROOT@snap1	0	-	31K	-
rpool/ROOT/s10zfsBE	4.78G	59.1G	4.76G	/
rpool/ROOT/s10zfsBE@snap1	15.6M	-	4.75G	-
rpool/dump	1.00G	59.1G	1.00G	-
rpool/dump@snap1	16K	-	1.00G	-
rpool/export	99K	59.1G	32K	/export
rpool/export@snap1	18K	-	32K	-
rpool/export/home	49K	59.1G	31K	/export/home
rpool/export/home@snap1	18K	-	31K	-
rpool/swap	2.06G	61.2G	16K	-
rpool/swap@snap1	0	-	16K	-

1 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 pool raiz.

```
# zfs rollback rpool@snap1  
# zfs rollback rpool/ROOT@snap1  
# zfs rollback rpool/ROOT/s10zfsBE@snap1
```

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 contém as seguintes seções:

- “Gerenciando de sistemas de arquivos ZFS (visão geral)” na página 175
- “Criando, destruindo e renomeando sistemas de arquivos ZFS” na página 176
- “Introduzindo as propriedades do ZFS” na página 179
- “Consultando informações sobre o sistema de arquivos ZFS” na página 192
- “Gerenciando propriedades do ZFS” na página 194
- “Montando sistemas de arquivos ZFS” na página 200
- “Compartilhando e descompartilhando sistemas de arquivos ZFS” na página 204
- “Definindo cotas e reservas do ZFS” na página 206
- “Atualização de sistemas de arquivos ZFS” na página 212

Gerenciando de sistemas de arquivos ZFS (visão geral)

Um sistema de arquivos ZFS está construído sobre um pool 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 6, “Trabalhando com instantâneos e clones do Oracle Solaris ZFS”](#). Para obter informações detalhadas sobre volumes ZFS, consulte [“Volumes ZFS” na página 275](#).

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 176
- “Destruindo um sistema de arquivos ZFS” na página 177
- “Renomeando um sistema de arquivos ZFS” na página 178

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 pool como se segue:

nome-do-pool/[nome-do-sistema-de-arquivos/]nome-do-sistema-de-arquivos

O nome do pool 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 30.

No exemplo abaixo, um sistema de arquivos denominado `jeff` é criado no sistema de arquivos `tank/home`.

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

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`. Neste exemplo, o sistema de arquivos `jeff` recém-criado está montado em `/tank/home/jeff`. Para mais informações sobre os pontos de montagem gerenciados automaticamente, consulte “[Gerenciando pontos de montagem do ZFS](#)” na página 200.

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 obter mais informações sobre propriedades do sistema de arquivos, consulte [“Introduzindo as propriedades do ZFS” na página 179](#).

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 201](#).

No exemplo a seguir, o sistema de arquivos `tank/home/mark` é destruído:

```
# zfs destroy tank/home/mark
```



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/matt
cannot unmount 'tank/home/matt': Device busy
```

```
# zfs destroy -f tank/home/matt
```

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/jeff
tank/ws/bill
tank/ws/mark
# 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/eric
cannot destroy 'tank/home/eric': filesystem has dependent clones
use '-R' to destroy the following datasets:
tank//home/eric-clone
# zfs destroy -R tank/home/eric
```



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 6, “Trabalhando com instantâneos e clones do Oracle Solaris ZFS”](#).

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.
- Realoque 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 `eric` para `eric_old`:

```
# zfs rename tank/home/eric tank/home/eric_old
```

O exemplo a seguir mostra como utilizar o `zfs rename` para realocar um sistema de arquivos:

```
# zfs rename tank/home/mark tank/ws/mark
```

Neste exemplo, o sistema de arquivos `mark` 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 pool e possuir espaço em disco suficiente para conter esse novo sistema de arquivos. Se o novo local não tiver espaço em disco suficiente, possivelmente por ter ultrapassado sua cota, a operação renomear falhará.

Para mais informações sobre as cotas, consulte [“Definindo cotas e reservas do ZFS” na página 206](#).

A operação `renomear` tenta uma sequência de desmontagem/remontagem no sistema de arquivos e nos sistemas de arquivos descendentes. O comando `renomear` 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 216](#).

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 informado ao contrário, as propriedades definidas nesta seção se aplicam a todos os tipos de conjunto de dados.

- [“Propriedades nativas somente leitura do ZFS” na página 187](#)
- [“Propriedades nativas definíveis do ZFS” na página 188](#)
- [“Propriedades de usuário do ZFS” na página 191](#)

As propriedades estão divididas em dois tipos: propriedades nativas e propriedades definidas pelo usuário. As propriedades nativas especificam 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 191](#).

A maioria das propriedades configuráveis também são herdáveis. Uma propriedade herdável é uma propriedade que, quando configurada no sistema de arquivos 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:

<code>local</code>	Indique que a propriedade foi definida explicitamente no conjunto de dados utilizando o comando <code>zfs set</code> conforme descrito em “Definindo propriedades do ZFS” na página 195 .
<code>inherited from <i>dataset-name</i></code>	Indique que a propriedade foi herdada do predecessor nomeado.
<code>default</code>	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 <code>local</code> .

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 191](#).

TABELA 5–1 Descrições de propriedades nativas do ZFS

Nome da Propriedade	Tipo	Valor Padrão	Descrição
aclinherit	Sequência de caracteres	secure	Controla como as entradas ACL são herdadas quando os arquivos e diretórios são criados. Os valores são discard, noallow, secure e passthrough. Para obter uma descrição desses valores, consulte “Propriedades da ACL” na página 241 .
aclmode	Sequência de caracteres	groupmask	Controla como uma entrada de ACL é modificada durante uma operação de chmod. Os valores são discard, groupmask e passthrough. Para obter uma descrição desses valores, consulte “Propriedades da ACL” na página 241 .
atime	Booleano	on	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.
available	Número	N/D	<p>A propriedade somente leitura que identifica a quantidade de espaço em disco disponível no sistema de dados e todos os seus filhos, mas não assume nenhuma outra atividade no pool. Como o espaço em disco é compartilhado dentro de um pool, o espaço disponível pode ser limitado por vários fatores, inclusive tamanho físico do pool, cotas, reservas ou outros conjuntos de dados dentro do pool.</p> <p>A abreviatura da propriedade é avail.</p> <p>Para mais informações sobre contabilidade de espaço em disco, consulte “Contabilidade de espaço em disco do ZFS” na página 32.</p>
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>Para obter mais informações, consulte “A propriedade canmount” na página 190.</p>

TABELA 5-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
soma de verificação	Sequência de caracteres	on	Controla a soma de verificação usada para verificar a integridade dos dados. O valor padrão é on, que seleciona automaticamente um algoritmo apropriado, atualmente fletcher4. Os valores são on, off, fletcher2, fletcher4 e sha256. O valor de off desativa a verificação da integridade dos dados do usuário. Um valor off não é recomendado.
compression	Sequência de caracteres	off	<p>Ativa ou desativa a compactação de um conjunto de dados. Os valores são on, off, lzjb, gzip e gzip-N. Nesse momento, definir esta propriedade como lzjb, gzip ou gzip-N tem o mesmo efeito que defini-la como on. 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 compress.</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. É possível ativar a compactação com o comando <code>zfs set compression=on dataset</code>.</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>
copies	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.
creation	Sequência de caracteres	N/D	A propriedade de somente leitura identifica a data e a hora em que um conjunto de dados foi criado.
devices	Booleano	on	Controla se os arquivos de dispositivo de um sistema de arquivos podem ser abertos.

TABELA 5-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
exec	Booleano	on	Controle se os programas dentro de um sistema de arquivos possuem permissão para serem executados. Também, quando definido como off, as chamadas de mmap(2) com PROT_EXEC não são permitidas.
mounted	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 sim ou não.
mountpoint	Sequência de caracteres	N/D	<p>Controla o ponto de montagem usado neste sistema de arquivos. Quando a propriedade mountpoint é alterada para um sistema de arquivos, o sistema de arquivos e os descendentes que herdam o ponto de montagem são desmontados. Se o novo valor for legacy, então eles permanecem desmontados. Do contrário, serão remontados automaticamente no novo local se a propriedade anterior era legacy ou none 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.</p> <p>Para obter mais informações sobre o uso desta propriedade, consulte “Gerenciando pontos de montagem do ZFS” na página 200.</p>
primarycache	Sequência de caracteres	tudo	Controle o que é armazenado no cache primário (ARC). Os valores possíveis são 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. 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. Você deve determinar se a configuração das propriedades de cache são adequadas para o seu ambiente.

TABELA 5-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
origin	Sequência de caracteres	N/D	<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 -r ou -f) enquanto o clone existir.</p> <p>Os sistemas de arquivo não clonados têm uma origem de none.</p>
quota	Número (ou none)	none	<p>Limita a quantidade de espaço em disco que um sistema 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 207.</p>
readonly	Booleano	off	<p>Controla se um conjunto de dados pode ser modificado. Ao definir como on, nenhuma modificação pode ser feita.</p> <p>A abreviatura da propriedade é <code>rdonly</code>.</p>
recordsize	Número	128K	<p>Especifica um tamanho de bloco sugerido para arquivos de um sistema de arquivos.</p> <p>A abreviatura da propriedade é <code>rcsize</code>. Para obter uma descrição detalhada, consulte “A propriedade recordsize” na página 190.</p>
referenced	Número	N/D	<p>Propriedade somente leitura que identifica a quantidade de dados acessível por um conjunto de dados, que podem ou não ser compartilhados com outros conjuntos de dados do pool.</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 <code>refer</code>.</p>

TABELA 5-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
refquota	Número (ou none)	none	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 rígido não inclui o espaço em disco utilizado pelos descendentes, como instantâneos e clones.
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. 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 <code>refreservation</code>. As reservas de <code>refreservation</code> 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 <code>refreservation</code> estiver configurado, um instantâneo será permitido somente se houver disponível espaço livre suficiente no pool fora da reserva para alojar o número atual de bytes <i>referenciados</i> no conjunto de dados.</p> <p>A abreviatura da propriedade é <code>refreserv</code>.</p>
reservation	Número (ou none)	none	<p>Configure a quantidade mínima de espaço em disco garantida para um sistema de arquivos e seus descendentes. Quando a quantidade de espaço em disco utilizada está abaixo desse valor, o sistema de arquivos é tratado como se estivesse utilizando a quantidade de espaço especificada por sua reserva. As reservas são consideradas no espaço de disco do sistema de arquivos pai usado e contadas com relação às cotas e reservas do sistema de arquivos pai.</p> <p>A abreviatura da propriedade é <code>reserv</code>.</p> <p>Para obter mais informações, consulte “Definindo reservas nos sistemas de arquivos ZFS” na página 210.</p>
secondarycache	Sequência de caracteres	tudo	Controla o que é armazenado no cache secundário (L2ARC). Os valores possíveis são <code>all</code> , <code>none</code> e <code>metadata</code> . Se forem definidas como <code>todos</code> , os dados do usuário e os metadados serão armazenados em cache. Se forem definidas como <code>nenhum</code> , nem os dados do usuário nem os metadados serão armazenados em cache. Se forem definidas como <code>metadados</code> , somente os metadados serão armazenados em cache.

TABELA 5-1 Descrições de propriedades nativas do ZFS (Continuação)

Nome da Propriedade	Tipo	Valor Padrão	Descrição
setuid	Booleano	on	Controla se o bit setuid é respeitado nos sistemas de arquivos.
shareiscsi	Sequência de caracteres	off	Controla se um volume ZFS está compartilhado como um destino iSCSI. Os valores das propriedades são on, off e type=disk. Se você deseja configurar o shareiscsi=on 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.
sharenfs	Sequência de caracteres	off	Controla se o sistema de arquivos está disponível no NFS e quais opções são utilizadas. Se for definido como on, o comando zfs share é chamado sem opções. Do contrário, o comando zfs share é chamado com opções equivalentes ao conteúdo desta propriedade. Se for definido como off, o sistema de arquivos é gerenciado com os comandos share e unshare de legado e o arquivo dfstab. Para mais informações sobre compartilhamento de sistemas de arquivos ZFS, consulte “Compartilhando e descompartilhando sistemas de arquivos ZFS” na página 204.
snapdir	Sequência de caracteres	hidden	Controla se o diretório .zfs está oculto ou visível na raiz do sistema de arquivos. Para obter mais informações sobre o uso de instantâneos, consulte “Visão geral dos instantâneos do ZFS” na página 213.
type	Sequência de caracteres	N/D	Propriedade somente leitura que identifica o tipo de conjunto de dados como filesystem (sistema de arquivos ou clone), volume ou snapshot.
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 obter uma descrição detalhada, consulte “A propriedade used” na página 188.
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 usedchild.

TABELA 5-1 Descrições de propriedades nativas do ZFS *(Continuação)*

Nome da Propriedade	Tipo	Valor Padrão	Descrição
<code>usedbydataset</code>	Número	<code>off</code>	A propriedade de somente leitura identifica a quantidade de espaço em disco utilizada por esse próprio conjunto de dados, que seria liberado se 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> .
<code>usedbyreservation</code>	Número	<code>off</code>	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>usedrefreserv</code> .
<code>usedbysnapshots</code>	Número	<code>off</code>	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> .
<code>version</code>	Número	N/D	Identifique a versão em disco do sistema de arquivos, que é independente da versão do pool. 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 obter uma descrição detalhada, consulte “A propriedade <code>volsize</code>” na página 191 .
<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> .

TABELA 5-1 Descrições de propriedades nativas do ZFS *(Continuação)*

Nome da Propriedade	Tipo	Valor Padrão	Descrição
zoned	Booleano	N/D	Indica se um sistema de arquivos foi adicionado à zona não global. Se esta propriedade estiver definida, o ponto de montagem não será respeitado na zona global e o ZFS não poderá montar tal sistema de arquivos quando solicitado. Quando uma zona é instalada, essa propriedade é definida para qualquer sistema de arquivos adicionado. Para obter mais informações sobre o uso do ZFS com zonas instaladas, consulte “Usando ZFS em um sistema Solaris com zonas instaladas” na página 278.
xattr	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 5-1](#).

As propriedades nativas de somente leitura estão listadas aqui e são descritas em [Tabela 5-1](#).

- available
- compressratio
- creation
- mounted
- origin
- referenced
- type
- used

Para obter informações detalhadas, consulte [“A propriedade used” na página 188.](#)

- usedbychildren
- usedbydataset
- usedbyrefreservation
- 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 32](#).

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 33](#).

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 5–1](#).

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 195](#) ou usando o comando `zfs create` conforme descrito em [“Criando um sistema de arquivos ZFS” na página 176](#). 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 206](#).

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 5–1. Se não for especificamente mencionado, um 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 5–1](#).

- `aclinherit`
Para uma descrição mais detalhada, consulte [“Propriedades da ACL” na página 241](#).
- `aclmode`
Para uma descrição mais detalhada, consulte [“Propriedades da ACL” na página 241](#).
- `atime`
- `canmount`
- `soma de verificação`
- `compression`
- `copies`
- `devices`
- `exec`
- `mountpoint`
- `primarycache`
- `quota`
- `readonly`
- `recordsize`
Para obter uma descrição detalhada, consulte [“A propriedade recordsize” na página 190](#).
- `refquota`
- `refreservation`
- `reservation`
- `secondarycache`
- `shareiscsi`
- `setuid`
- `snapdir`
- `version`
- `volsize`
Para obter uma descrição detalhada, consulte [“A propriedade volsize” na página 191](#).
- `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 sistema de arquivos 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
```

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 esparsa especificando o sinalizador `-s` para `zfs create -V` ou alterando a reserva depois que o volume for criado. Um *volume esparsa* é um volume em que a reserva não é igual ao tamanho do volume. Em um volume esparsa, as alterações para `volsize` não se refletem na reserva.

Para obter mais informações sobre o uso de volumes, consulte “[Volumes ZFS](#)” na página 275.

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 utilizadas para anotar conjuntos de dados com informações significativas no ambiente.

Os nomes das propriedades do usuário devem estar de acordo com as seguintes convenções:

- 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.oracle.` estão reservados para utilização da Oracle Corporation.

Os valores da propriedade do usuário devem estar de acordo com as seguintes convenções:

- Eles devem consistir de sequências 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 179](#).

Por exemplo:

```
# zfs list
users          2.00G 64.9G 32K /users
users/home     2.00G 64.9G 35K /users/home
users/home/cindy 548K 64.9G 548K /users/home/cindy
users/home/mark 1.00G 64.9G 1.00G /users/home/mark
users/home/neil 1.00G 64.9G 1.00G /users/home/neil
```

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 -t all -r users/home/mark
NAME                USED  AVAIL  REFER  MOUNTPOINT
users/home/mark      1.00G  64.9G  1.00G  /users/home/mark
users/home/mark@yesterday    0      -  1.00G  -
users/home/mark@today        0      -  1.00G  -
```

É possível utilizar o comando `zfs list` com o ponto de montagem de um sistema de arquivos. Por exemplo:

```
# zfs list /user/home/mark
NAME                USED  AVAIL  REFER  MOUNTPOINT
users/home/mark     1.00G  64.9G  1.00G  /users/home/mark
```

O exemplo a seguir ilustra como exibir informações básicas sobre `tank/home/gina` e todos os seus conjuntos de dados descendentes:

```
# zfs list -r users/home/gina
NAME                USED  AVAIL  REFER  MOUNTPOINT
users/home/gina      2.00G  62.9G   32K  /users/home/gina
users/home/gina/projects  2.00G  62.9G   33K  /users/home/gina/projects
users/home/gina/projects/fs1  1.00G  62.9G  1.00G  /users/home/gina/projects/fs1
users/home/gina/projects/fs2  1.00G  62.9G  1.00G  /users/home/gina/projects/fs2
```

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 179](#). 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 `sharenamefs` e `mountpoint`.

```
# zfs list -r -o name,sharenamefs,mountpoint users/home
NAME                SHARENFS  MOUNTPOINT
users/home           on        /users/home
users/home/cindy     on        /users/home/cindy
users/home/gina      on        /users/home/gina
users/home/gina/projects  on        /users/home/gina/projects
users/home/gina/projects/fs1  on        /users/home/gina/projects/fs1
users/home/gina/projects/fs2  on        /users/home/gina/projects/fs2
users/home/mark      on        /users/home/mark
users/home/neil      on        /users/home/neil
```

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 5-2 Tipos de Objetos ZFS

Tipo	Descrição
filesystem	Sistemas de arquivos e clones
volume	Volumes
share	Compartilhamento de sistema de arquivos
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 -r -t filesystem -o name,used users/home
NAME                                USED
users/home                          4.00G
users/home/cindy                     548K
users/home/gina                      2.00G
users/home/gina/projects             2.00G
users/home/gina/projects/fs1         1.00G
users/home/gina/projects/fs2         1.00G
users/home/mark                      1.00G
users/home/neil                      1.00G
```

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 -r -H -o name users/home
users/home
users/home/cindy
users/home/gina
users/home/gina/projects
users/home/gina/projects/fs1
users/home/gina/projects/fs2
users/home/mark
users/home/neil
```

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 195
- “Herdando propriedades do ZFS” na página 195
- “Consultando propriedades do ZFS” na página 196

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 188.

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 tamanhos crescentes): `BKMGTP EZ`. 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 20 GB no sistema de arquivos `users/home/mark`:

```
# zfs set quota=20G users/home/mark
# zfs set quota=20g users/home/mark
# zfs set quota=20GB users/home/mark
# zfs set quota=20gb users/home/mark
```

Se você tentar definir uma propriedade em um sistema de arquivos que esteja 100% cheio, verá uma mensagem semelhante a esta:

```
# zfs set quota=20gb users/home/mark
cannot set property for '/users/home/mark': out of space
```

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 das cotas e reservas, herdam seu valor do sistema de arquivos pai, a menos que uma cota ou reserva seja definida explicitamente no

sistema de arquivos 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 a seguir usa o comando `zfs set` para ativar a compactação do sistema de arquivos `tank/home/jeff`. 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/jeff
# zfs get -r compression tank/home
NAME                PROPERTY  VALUE    SOURCE
tank/home            compression off       default
tank/home/eric       compression off       default
tank/home/eric@today  compression -         -
tank/home/jeff        compression on        local
# zfs inherit compression tank/home/jeff
# zfs get -r compression tank/home
NAME                PROPERTY  VALUE    SOURCE
tank/home            compression off       default
tank/home/eric       compression off       default
tank/home/eric@today  compression -         -
tank/home/jeff        compression off       default
```

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 sistema de arquivos 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 192](#). 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 5-3 Possíveis valores de SOURCE (comando zfs get)

Valor de origem	Descrição
default	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.
inherited from <i>dataset-name</i>	Esse valor de propriedade é herdado a partir do conjunto de dados pai especificado em <i>dataset-name</i> .
local	Este valor de propriedade foi explicitamente definido para este conjunto de dados usando zfs set.
temporary	Esse valor de propriedade foi definido utilizando a opção zfs mount -o 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 203 .
-(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:

Observação – As propriedades casesensitivity, nbmand, normalization, sharesmb, utf8only e vscan 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.

```
# zfs get all tank/home
NAME      PROPERTY  VALUE      SOURCE
tank/home type      filesystem -
tank/home creation Mon Dec 3 13:10 2012 -
tank/home used      291K      -
tank/home available 58.7G     -
tank/home referenced 291K      -
tank/home compressratio 1.00x     -
tank/home mounted   yes       -
tank/home quota     none      default
tank/home reservation none      default
tank/home recordsize 128K     default
tank/home mountpoint /tank/home default
tank/home sharenfs   off       default
tank/home checksum  on        default
tank/home compression off       default
```

tank/home	atime	on	default
tank/home	devices	on	default
tank/home	exec	on	default
tank/home	setuid	on	default
tank/home	readonly	off	default
tank/home	zoned	off	default
tank/home	snappdir	hidden	default
tank/home	aclmode	discard	default
tank/home	aclinherit	restricted	default
tank/home	canmount	on	default
tank/home	shareiscsi	off	default
tank/home	xattr	on	default
tank/home	copies	1	default
tank/home	version	5	-
tank/home	utf8only	off	-
tank/home	normalization	none	-
tank/home	casesensitivity	mixed	-
tank/home	vscan	off	default
tank/home	nbmand	off	default
tank/home	sharesmb	off	default
tank/home	refquota	none	default
tank/home	refreservation	none	default
tank/home	primarycache	all	default
tank/home	secondarycache	all	default
tank/home	usedbysnapshots	0	-
tank/home	usedbydataset	291K	-
tank/home	usedbychildren	0	-
tank/home	usedbyrefreservation	0	-
tank/home	logbias	latency	default
tank/home	sync	standard	default
tank/home	rekeydate	-	default
tank/home	rstchown	on	default

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 a seguir ilustra todas as propriedades definidas localmente em `tank/ws`.

```
# zfs get -s local all tank/ws
NAME      PROPERTY      VALUE      SOURCE
tank/ws    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 sistema de arquivos especificado. No exemplo a seguir, todas as propriedades temporárias de todos os sistemas de arquivos dentro de `tank/home` são exibidas repetidamente:

```
# zfs get -r -s temporary all tank/home
NAME      PROPERTY      VALUE      SOURCE
tank/home    atime         off        temporary
tank/home/jeff atime         off        temporary
tank/home/mark quota         20G       temporary
```

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/jeff           atime                off                  local
tank/home/mark           quota                20G                  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 179.
- 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 sistema de arquivos 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
```

Montando sistemas de arquivos ZFS

Esta seção descreve como o ZFS monta e compartilha sistemas de arquivos.

- “Gerenciando pontos de montagem do ZFS” na página 200
- “Montando sistemas de arquivos ZFS” na página 202
- “Usando propriedades de montagem temporárias” na página 203
- “Desmontando sistemas de arquivos” na página 204

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 sistema de arquivos de um pool no momento da criação utilizando, de `zpool create`, a opção `-m`. Para obter mais informações sobre a criação de pools de armazenamento, consulte “[Criando Pools de Armazenamento do ZFS](#)” na página 50.

Todos os sistemas de arquivos ZFS são montados pelo ZFS no momento da inicialização utilizando o serviço `svc://system/filesystem/local` do SMF (Service Management Facility). Os sistemas de arquivos são montados em `/path`, onde `path` é o nome do sistema de arquivos.

É possível substituir 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.

Os sistemas de arquivos ZFS são montados automaticamente no tempo de inicialização sem exigir a edição do arquivo `/etc/vfstab`.

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 190.

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 legados](#)” na página 201.

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 sistema de arquivos cuja propriedade mountpoint não for legacy é gerenciado pelo ZFS. No exemplo a seguir, o sistema de arquivos 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 legados

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 sistemas de arquivos desse tipo. Os exemplos abaixo ilustram como configurar e gerenciar um sistema de arquivos ZFS no modo legado:

```
# zfs set mountpoint=legacy tank/home/eric
# 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/eric	-	/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 26](#).

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 | grep tank/home
zfs mount | grep tank/home
tank/home                /tank/home
tank/home/jeff           /tank/home/jeff
```

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. Por exemplo:

```
# zfs mount tank/home/lori
cannot mount 'tank/home/lori': filesystem already mounted
```

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 tank/home/bill
cannot mount 'tank/home/bill': 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 sistema de arquivos. A correlação entre propriedades e opções de montagem é a seguinte:

TABELA 5-4 Propriedades relacionadas à montagem ZFS e opções de montagem

Propriedade	Opções de montagem
atime	atime/noatime
devices	devices/nodevices
exec	exec/noexec
nbmand	nbmand/nonbmand
readonly	ro/rw
setuid	setuid/nosetuid
xattr	xattr/noaxttr

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 sistema de arquivos 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 users/home/neil
```

Para alterar temporariamente um valor de propriedade no sistema de arquivos que esteja atualmente montado, utilize a opção `remount` especial. No exemplo abaixo, a propriedade `atime` é temporariamente alterada para `off` para um sistema de arquivos atualmente montado:

```
# zfs mount -o remount,noatime users/home/neil
NAME                PROPERTY  VALUE  SOURCE
users/home/neil     atime     off    temporary
# zfs get atime users/home/perrin
```

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 `umount` 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 users/home/mark
```

No exemplo a seguir, o sistemas de arquivos é desmontado pelo seu ponto de montagem:

```
# zfs unmount /users/home/mark
```

O comando `umount` 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/eric
cannot unmount '/tank/home/eric': Device busy
# zfs unmount -f tank/home/eric
```

Para proporcionar compatibilidade com versões anteriores, o comando `umount` de legado pode ser utilizado para desmontar sistemas de arquivos ZFS. Por exemplo:

```
# umount /tank/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/eric
```

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/bill
# zfs create tank/home/mark
# zfs set sharenfs=ro tank/home/bob
```

`tank/home/bill` e `tank/home/mark` são inicialmente compartilhados como graváveis porque herdam a propriedade `sharenfs` de `tank/home`. Depois que a propriedade é configurada como `ro` (somente leitura), `tank/home/mark` é compartilhado como somente leitura independente da propriedade `sharenfs` que é configurada como `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/mark
```

Este comando descompartilha o sistema de arquivos `tank/home/mark`. 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/mark
```

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 sistema de arquivos em que são definidas e todos os descendentes desse sistema de arquivos.

Isto é, se uma cota é definida no sistema de arquivos `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 dada uma reserva, `tank/home` e *todos os seus descendentes* utilizarão tal reserva. A quantidade de espaço em disco utilizado por um sistema de arquivos 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 sistema de arquivos e seus descendentes.
- As propriedades `refquota` e `refreservation` são apropriadas para gerenciar o espaço em disco consumido pelo sistema de arquivos.

- 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 207](#) and [“Definindo reservas nos sistemas de arquivos ZFS” na página 210](#).

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/jeff`:

```
# zfs set quota=10G tank/home/jeff
# zfs get quota tank/home/jeff
NAME                PROPERTY  VALUE   SOURCE
tank/home/jeff      quota     10G     local
```

As cotas também afetam a saída dos comandos `zfs list` e `df`. Por exemplo:

```
# zfs list -r tank/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home           1.45M  66.9G   36K    /tank/home
tank/home/eric      547K   66.9G  547K    /tank/home/eric
tank/home/jeff      322K   10.0G  291K    /tank/home/jeff
tank/home/jeff/ws    31K   10.0G   31K    /tank/home/jeff/ws
tank/home/lori      547K   66.9G  547K    /tank/home/lori
tank/home/mark      31K   66.9G   31K    /tank/home/mark
# df -h /tank/home/jeff
Filesystem           Size  Used Avail Use% Mounted on
tank/home/jeff       10G   306K   10G    1% /tank/home/jeff
```

Observe que, embora `tank/home` possua 66,9 GB de espaço em disco disponível, `tank/home/jeff` e `tank/home/jeff/ws` possuem, cada um, somente 10 GB de espaço em disco disponível, devido à cota em `tank/home/jeff`.

Não é possível definir uma cota para uma quantidade menor do que a atualmente em uso por um sistema de arquivos. Por exemplo:

```
# zfs set quota=10K tank/home/jeff
cannot set property for 'tank/home/jeff':
size is less than current used or reserved space
```

É possível definir uma `refquota` em um sistema de arquivos que limite a quantidade de espaço em disco que o sistema de arquivos pode consumir. Esse limite de disco rígido não inclui o espaço em disco que é utilizado pelos descendentes. Por exemplo, a cota de 10 GB do aluno A não é afetada pelo espaço consumido por instantâneos.

```
# zfs set refquota=10g students/studentA
# zfs list -t all -r students
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
students	150M	66.8G	32K	/students
students/studentA	150M	9.85G	150M	/students/studentA
students/studentA@yesterday	0	-	150M	-

```
# zfs snapshot students/studentA@today
# zfs list -t all -r students
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
students	150M	66.8G	32K	/students
students/studentA	150M	9.90G	100M	/students/studentA
students/studentA@yesterday	50.0M	-	150M	-
students/studentA@today	0	-	100M	-

Para maior comodidade, é possível configurar outra cota no sistema de arquivos para ajudar a gerenciar o espaço em disco que é utilizado pelos instantâneos. Por exemplo:

```
# zfs set quota=20g students/studentA
# zfs list -t all -r students
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
students	150M	66.8G	32K	/students
students/studentA	150M	9.90G	100M	/students/studentA
students/studentA@yesterday	50.0M	-	150M	-
students/studentA@today	0	-	100M	-

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@labstaff=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@labstaff students/labstaff
NAME                PROPERTY          VALUE          SOURCE
students/labstaff    groupquota@labstaff 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    350M  none
POSIX User  student1 426M  10G
# zfs groupspace students/labstaff
TYPE    NAME    USED  QUOTA
POSIX Group  labstaff 250M  20G
POSIX Group  root    350M  none
```

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 550M           local
# zfs get groupused@labstaff students/labstaff
NAME                PROPERTY          VALUE          SOURCE
students/labstaff    groupused@labstaff 250            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@student1=none students/compsci
# zfs set groupquota@labstaff=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      350M none
POSIX User student1 550M  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      350M none
POSIX User student1 550M  10G
# quota student1
# quota -v student1
Disk quotas for student1 (uid 102):
Filesystem      usage  quota  limit  timeleft  files  quota  limit  timeleft
/students/compsci
                563287 10485760 10485760          -          -          -          -
```

Definindo reservas nos sistemas de arquivos ZFS

Uma *reserva* ZFS é uma alocação do espaço em disco do pool 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 estiver atualmente disponível no pool. A quantidade total de reservas livres e não consumidas não pode exceder a quantidade de espaço em disco não utilizado no pool. A reservas do ZFS podem ser definidas e exibidas com os comandos `zfs set` e `zfs get`. Por exemplo:

```
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
NAME      PROPERTY  VALUE  SOURCE
tank/home/bill reservation 5G      local
```

As reservas podem afetar a saída do comando `zfs list`. Por exemplo:

```
# zfs list -r tank/home
NAME      USED  AVAIL  REFER  MOUNTPOINT
tank/home 5.00G 61.9G  37K    /tank/home
```

tank/home/bill	31K	66.9G	31K	/tank/home/bill
tank/home/jeff	337K	10.0G	306K	/tank/home/jeff
tank/home/lori	547K	61.9G	547K	/tank/home/lori
tank/home/mark	31K	61.9G	31K	/tank/home/mark

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/bill. As reservas são consideradas no cálculo de espaço em disco utilizado do sistema de arquivos 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 pool 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/bill
# zfs set reservation=5G tank/home/bill
# zfs get reservation tank/home/bill
```

NAME	PROPERTY	VALUE	SOURCE
tank/home/bill	reservation	5G	local

É possível definir uma reserva refreservation 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,refreservation 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 pool não reservado suficiente existe fora da reserva para acomodar o número atual de bytes *referenciados* no conjunto de dados.

Atualização de sistemas de arquivos ZFS

Se você tiver sistemas de arquivos ZFS de uma versão anterior do Solaris, pode atualizar seus sistemas de arquivos com o comando `zfs upgrade` para utilizar os recursos do sistema de arquivos na versão atual. Além disso, esse comando notifica você quando seus sistema estão executando versões mais antigas.

Por exemplo, este sistema de arquivos está na versão atual 5.

```
# zfs upgrade
This system is currently running ZFS filesystem version 5.
```

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

Use este comando para identificar os recursos que estão disponíveis com cada versão do sistema de arquivos.

```
# zfs upgrade -v
The following filesystem versions are supported:

VER  DESCRIPTION
---  -
1    Initial ZFS filesystem version
2    Enhanced directory entries
3    Case insensitive and File system unique identifier (FUID)
4    userquota, groupquota properties
5    System attributes
```

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

Trabalhando com instantâneos e clones do Oracle Solaris ZFS

Este capítulo descreve como criar e gerenciar instantâneos e clones do Oracle Solaris ZFS. Também são fornecidas informações sobre como salvar instantâneos.

Este capítulo traz as seguintes seções:

- “Visão geral dos instantâneos do ZFS” na página 213
- “Criando e destruindo instantâneos do ZFS” na página 214
- “Exibindo e acessando instantâneos do ZFS” na página 217
- “Retornando um instantâneo ZFS” na página 218
- “Visão geral dos clones do ZFS” na página 220
- “Criando um clone do ZFS” na página 221
- “Destruindo um clone do ZFS” na página 221
- “Substituindo um sistema de arquivos ZFS por um clone do ZFS” na página 222
- “Enviando e recebendo dados do ZFS” na página 223

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 consomem espaço em disco adicional dentro do pool. No entanto, à medida que os dados do conjunto de dados ativo são alterados, o instantâneo consome espaço em disco, continuando a fazer referência aos dados antigos, o que impede que o espaço em disco seja liberado.

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 pool 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 223](#).

- [“Criando e destruindo instantâneos do ZFS” na página 214](#)
- [“Exibindo e acessando instantâneos do ZFS” na página 217](#)
- [“Retornando um instantâneo ZFS” na página 218](#)

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 30](#).

No exemplo seguinte, é criado um instantâneo de `tank/home/cindy` denominado `friday`.

```
# zfs snapshot tank/home/cindy@friday
```

É possível criar instantâneos para todos os sistemas de arquivos descendentes com a opção `-r`. Por exemplo:

```
# zfs snapshot -r tank/home@snap1
# zfs list -t snapshot -r tank/home
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank/home@snap1	0	-	2.11G	-
tank/home/cindy@snap1	0	-	115M	-
tank/home/lori@snap1	0	-	2.00G	-
tank/home/mark@snap1	0	-	2.00G	-
tank/home/tim@snap1	0	-	57.3M	-

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/cindy@friday
cannot set property for 'tank/home/cindy@friday':
this property can not be modified for snapshots
```

Os instantâneos são destruídos com o comando `zfs destroy`. Por exemplo:

```
# zfs destroy tank/home/cindy@friday
```

Um conjunto de dados não pode ser destruído se existirem instantâneos deste conjunto de dados. Por exemplo:

```
# zfs destroy tank/home/cindy
cannot destroy 'tank/home/cindy': filesystem has children
use '-r' to destroy the following datasets:
tank/home/cindy@tuesday
tank/home/cindy@wednesday
tank/home/cindy@thursday
```

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 177](#).

Mantendo instantâneos do ZFS

Se você possui uma política de instantâneo automático diferente, na qual instantâneos mais antigos estão sendo destruídos sem aviso pelo `zfs` porque não existem 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 um instantâneo com clones a ser excluídos aguardando a remoção do último clone ao utilizar o comando `zfs destroy -d`. Cada instantâneo possui uma contagem associada para referência do usuário, inicializada no zero. Esta contagem aumenta de 1 em 1 quando uma contenção é colocada em um instantâneo e diminui de 1 em 1 quando uma contenção é liberada.

Na versão anterior do Oracle Solaris, instantâneos poderiam ser destruídos apenas através do comando `zfs destroy` se esses não tivessem clones. Nesta versão do Oracle 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 seguinte sintaxe coloca uma tag de contenção, `keep`, em `tank/home/cindy/snap@1`:

```
# zfs hold keep tank/home/cindy@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 tem sua própria tag de namespace, e as tags 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/cindy@snap1
cannot destroy 'tank/home/cindy@snap1': dataset is busy
```

Para destruir um instantâneo contido, utilize a opção `-d`. Por exemplo:

```
# zfs destroy -d tank/home/cindy@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   Fri Aug  3 15:15:53 2012

# zfs holds -r tank/home@now
NAME          TAG    TIMESTAMP
tank/home/cindy@now    keep   Fri Aug  3 15:15:53 2012
tank/home/lori@now     keep   Fri Aug  3 15:15:53 2012
tank/home/mark@now     keep   Fri Aug  3 15:15:53 2012
tank/home/tim@now      keep   Fri Aug  3 15:15:53 2012
tank/home@now          keep   Fri Aug  3 15:15:53 2012
```

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
```

Duas 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 `userrefs` 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 mesmo pool e conjunto de dados a partir do qual foram criados. Por exemplo:

```
# zfs rename tank/home/cindy@snap1 tank/home/cindy@today
```

Além disso, a seguinte sintaxe do atalho é equivalente à sintaxe anterior:

```
# zfs rename tank/home/cindy@snap1 today
```

A seguinte operação `rename` de instantâneo não é suportada porque o pool de destino e o nome do sistema de arquivos são diferentes do pool e do sistema de arquivos onde o instantâneo foi criado:


```
# zfs rename tank/home/cindy@today pool/home/cindy@aturday
cannot rename to 'pool/home/cindy@today': snapshots must be part of same
dataset
```

Como recurso, é possível renomear instantâneos com o comando `zfs rename -r`. Por exemplo:

```
# zfs list -t snapshot -r users/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
users/home@now       23.5K  -      35.5K  -
users/home@yesterday 0      -      38K    -
users/home/lori@yesterday 0      -      2.00G  -
users/home/mark@yesterday 0      -      1.00G  -
users/home/neil@yesterday 0      -      2.00G  -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -t snapshot -r users/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
users/home@now       23.5K  -      35.5K  -
users/home@2daysago 0      -      38K    -
users/home/lori@2daysago 0      -      2.00G  -
users/home/mark@2daysago 0      -      1.00G  -
users/home/neil@2daysago 0      -      2.00G  -
```

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 `ank/home/cindy` 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/cindy/.zfs/snapshot
thursday  tuesday  wednesday
```

Os instantâneos podem ser listados da seguinte forma:

```
# zfs list -t snapshot -r tank/home
NAME                USED  AVAIL  REFER  MOUNTPOINT
tank/home/cindy@tuesday 45K   -      2.11G  -
tank/home/cindy@wednesday 45K   -      2.11G  -
tank/home/cindy@thursday 0      -      2.17G  -
```

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/cindy@tuesday             Fri Aug 3 15:18 2012
tank/home/cindy@wednesday           Fri Aug 3 15:19 2012
tank/home/cindy@thursday            Fri Aug 3 15:19 2012
tank/home/lori@today                 Fri Aug 3 15:24 2012
tank/home/mark@today                 Fri Aug 3 15:24 2012
```

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.

Um valor de propriedade `referenced` do espaço do instantâneo é o mesmo que o do sistema de arquivos quando 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 -r rpool
NAME                                AVAIL    USED    USED SNAP    USED DS    USED REF RESERV    USED CHILD
rpool                               59.1G    7.84G    21K          109K        0                7.84G
rpool@snap1                          -        21K      -            -           -                -
rpool/ROOT                            59.1G    4.78G    0            31K         0                4.78G
rpool/ROOT@snap1                     -         0        -            -           -                -
rpool/ROOT/zfsBE                     59.1G    4.78G    15.6M        4.76G       0                0
rpool/ROOT/zfsBE@snap1               -       15.6M    -            -           -                -
rpool/dump                           59.1G    1.00G    16K          1.00G       0                0
rpool/dump@snap1                     -        16K      -            -           -                -
rpool/export                         59.1G    99K      18K          32K         0                49K
rpool/export@snap1                   -        18K      -            -           -                -
rpool/export/home                    59.1G    49K      18K          31K         0                0
rpool/export/home@snap1              -        18K      -            -           -                -
rpool/swap                           61.2G    2.06G    0            16K         2.06G           0
rpool/swap@snap1                     -         0        -            -           -                -
```

Para obter uma descrição dessas propriedades, consulte a [Tabela 5–1](#).

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/cindy` é revertido ao instantâneo `tuesday`:

```
# zfs rollback tank/home/cindy@tuesday
cannot rollback to 'tank/home/cindy@tuesday': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
tank/home/cindy@wednesday
tank/home/cindy@thursday
# zfs rollback -r tank/home/cindy@tuesday
```

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/cindy
NAME                                CREATION
tank/home/cindy@tuesday             Fri Aug  3 15:18 2012
```

Identificação de diferenças do instantâneo do ZFS (`zfs diff`)

Você pode determinar diferenças de instantâneos do ZFS usando o comando `zfs diff`.

Por exemplo, considere que estes dois instantâneos sejam criados:

```
$ ls /tank/home/tim
fileA
$ zfs snapshot tank/home/tim@snap1
$ ls /tank/home/tim
fileA fileB
$ zfs snapshot tank/home/tim@snap2
```

Por exemplo, para identificar as diferenças entre dois instantâneos, use sintaxe semelhante à seguinte:

```
$ zfs diff tank/home/tim@snap1 tank/home/tim@snap2
M      /tank/home/tim/
+      /tank/home/tim/fileB
```

Na saída, M indica que o diretório foi modificado. + indica que fileB existe no instantâneo posterior.

O R na saída a seguir indica que um arquivo em um instantâneo foi renomeado.

```
$ mv /tank/cindy/fileB /tank/cindy/fileC
$ zfs snapshot tank/cindy@snap2
$ zfs diff tank/cindy@snap1 tank/cindy@snap2
M      /tank/cindy/
R      /tank/cindy/fileB -> /tank/cindy/fileC
```

A tabela a seguir resume as alterações de arquivo ou diretório que são identificadas pelo comando `zfs diff`.

Alteração de arquivo ou diretório	Identificador
O arquivo ou diretório foi modificado ou o link do arquivo ou diretório foi alterado	M
O arquivo ou diretório está presente no instantâneo mais antigo, mas não no instantâneo mais recente	–
O arquivo ou diretório está presente no instantâneo mais recente, mas não no instantâneo mais antigo	+
O arquivo ou diretório foi renomeado	A

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

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, é possível realizar um 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 sistema 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 195](#).

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 221](#)
- [“Destruindo um clone do ZFS” na página 221](#)
- [“Substituindo um sistema de arquivos ZFS por um clone do ZFS” na página 222](#)

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 pool diferente daquele no qual se encontra o instantâneo do sistema de arquivos original.

No exemplo a seguir, um novo clone chamado `tank/home/matt/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/matt/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/matt/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 221](#).

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.

- [“Salvando dados do ZFS com outros produtos de backup” na página 224](#)
- [“Enviando um instantâneo do ZFS” na página 226](#)
- [“Recebendo um instantâneo do ZFS” na página 227](#)
- [“Aplicação de valores de propriedade diferentes a um fluxo de instantâneos do ZFS” na página 228](#)
- [“Enviando e recebendo fluxos complexos de instantâneos do ZFS” na página 231](#)
- [“Replicação remota de dados do ZFS” na página 233](#)

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 213](#).

- **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 produtos de backup do ZFS e de terceiros, consulte as Notas de Versão do Solaris 10.

Identificando fluxos de instantâneos do ZFS

Um instantâneo de um sistema de arquivos ou volume do ZFS é convertido em um fluxo de instantâneo usando o comando `zfs send`. Em seguida, é possível usar o fluxo de instantâneo para recriar um sistema de arquivos ou volume do ZFS usando o comando `zfs receive`.

Dependendo das opções de `zfs send` usadas para criar o fluxo de instantâneo, diferentes tipos de formatos de fluxo são gerados.

- Fluxo completo – Consiste em todo o conteúdo do conjunto de dados desde a criação do conjunto de dados até o instantâneo especificado.
O fluxo padrão gerado pelo comando `zfs send` é um fluxo completo. Ele contém um sistema de arquivos ou volume até, incluindo, o instantâneo especificado. O fluxo não contém instantâneos diferentes do instantâneo especificado na linha de comandos.
- Fluxo incremental – Consiste nas diferenças entre um instantâneo e outro instantâneo.

Um pacote de fluxos é um tipo de fluxo que contém um ou mais fluxos completos ou incrementais. Existem três tipos de pacotes de fluxos:

- Pacote de fluxos de replicação – Consiste no conjunto de dados especificado e em seus descendentes. Inclui todos os instantâneos intermediários. Se a origem de um conjunto de dados clonado não for um descendente do instantâneo especificado na linha de comandos, esse conjunto de dados de origem não será incluído no pacote de fluxos. Para receber o fluxo, o conjunto de dados de origem deve existir no pool de armazenamento de destino.

Considere a lista a seguir de conjuntos de dados e suas origens. Pressuponha que eles foram criados na ordem em que aparecem abaixo.

NAME	ORIGIN
pool/a	-
pool/a/1	-


```

pool/a/1@clone      -
pool/b              -
pool/b/1            pool/a/1@clone
pool/b/1@clone2     -
pool/b/2            pool/b/1@clone2
pool/b@pre-send     -
pool/b/1@pre-send   -
pool/b/2@pre-send   -
pool/b@send         -
pool/b/1@send       -
pool/b/2@send       -

```

Um pacote de fluxos de replicação foi criado com a seguinte sintaxe:

```
# zfs send -R pool/b@send ...
```

Consiste nos seguintes fluxos completos e incrementais:

TYPE	SNAPSHOT	INCREMENTAL FROM
full	pool/b@pre-send	-
incr	pool/b@send	pool/b@pre-send
incr	pool/b/1@clone2	pool/a/1@clone
incr	pool/b/1@pre-send	pool/b/1@clone2
incr	pool/b/1@send	pool/b/1@send
incr	pool/b/2@pre-send	pool/b/1@clone2
incr	pool/b/2@send	pool/b/2@pre-send

Na saída anterior, o instantâneo pool/a/1@clone não está incluído no pacote de fluxos de replicação. Assim, esse pacote de fluxos de replicação só pode ser recebido em um pool que já tenha o instantâneo pool/a/1@clone.

- Pacote de fluxos recursivo – Consiste no conjunto de dados especificado e em seus descendentes. Diferentemente dos pacotes de fluxo de replicação, os instantâneos intermediários não são incluídos, a não ser que sejam a origem de um conjunto de dados clonado incluído no fluxo. Por padrão, se a origem de um conjunto de dados não for um descendente do instantâneo especificado na linha de comandos, o comportamento será semelhante ao dos fluxos de replicação. No entanto, um fluxo recursivo independente, discutido abaixo, é criado de forma que não haja dependências externas.

Um pacote de fluxos recursivos foi criado com a seguinte sintaxe:

```
# zfs send -r pool/b@send ...
```

Consiste nos seguintes fluxos completos e incrementais:

TYPE	SNAPSHOT	INCREMENTAL FROM
full	pool/b@send	-
incr	pool/b/1@clone2	pool/a/1@clone
incr	pool/b/1@send	pool/b/1@clone2
incr	pool/b/2@send	pool/b/1@clone2

Na saída anterior, o instantâneo pool/a/1@clone não está incluído no pacote de fluxos recursivos. Como tal, esse pacote de fluxos recursivos só pode ser recebido em um pool que já tenha um instantâneo pool/a/1@clone. Esse comportamento é semelhante ao cenário de pacote de fluxos de replicação descrito acima.

- Pacote de fluxos recursivos independente - Não depende de conjuntos de dados que não estejam incluídos no pacote de fluxos. Esse pacote de fluxos recursivos é criado com a seguinte sintaxe:

```
# zfs send -rc pool/b@send ...
```

Consiste nos seguintes fluxos completos e incrementais:

TYPE	SNAPSHOT	INCREMENTAL FROM
full	pool/b@send	-
full	pool/b/1@clone2	
incr	pool/b/1@send	pool/b/1@clone2
incr	pool/b/2@send	pool/b/1@clone2

Observe que o fluxo recursivo independente tem um fluxo completo do instantâneo pool/b/1@clone2, tornando possível o recebimento do instantâneo pool/b/1 sem dependências externas.

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 pool do mesmo sistema ou em outro pool de outro sistema utilizado para armazenar dados de backup. Por exemplo, para enviar o fluxo de instantâneo de outro pool para o mesmo sistema, utilize uma sintaxe semelhante à seguinte:

```
# zfs send tank/dana@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:

```
sys1# zfs send tank/dana@snap1 | ssh sys2 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:

```
sys1# zfs send -i tank/dana@snap1 tank/dana@snap2 | ssh sys2 zfs recv newtank/dana
```

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.

Observação – O acesso às informações de arquivo no sistema de arquivos original recebido pode causar falha na operação incremental de recebimento de instantâneo com uma mensagem semelhante a:

```
cannot receive incremental stream of tank/dana@snap2 into newtank/dana:  
most recent snapshot of tank/dana@snap2 does not match incremental source
```

Considere configurar a propriedade `atime` como desativada se você precisar acessar as informações de arquivo no sistema de arquivos original recebido e se precisar receber instantâneos incrementais no sistema de arquivos.

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:

```
sys1# zfs send -i snap1 tank/dana@snap2 | ssh sys2 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:

```
sys2# 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:

```
sys1# zfs send -i tank/dana@snap2 tank/dana@snap3 | ssh sys2 zfs recv -F newtank/dana
```

Quando um instantâneo incremental é recebido, o sistema de arquivos de destino já dever 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:

```
sys1# zfs send -i tank/dana@snap4 tank/dana@snap5 | ssh sys2 zfs recv newtank/dana
cannot receive: destination has been modified since most recent snapshot
```

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 e 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
```

Aplicação de valores de propriedade diferentes a um fluxo de instantâneos do ZFS

Você pode enviar um fluxo de instantâneos com um determinado valor de propriedade do sistema de arquivos, mas pode especificar um valor de propriedade local quando o fluxo de instantâneos for recebido. Ou você pode especificar que o valor de propriedade original seja

usado quando o fluxo de instantâneos for recebido para recriar o sistema de arquivos original. Além disso, você pode desativar uma propriedade do sistema de arquivos quando o fluxo de instantâneos for recebido.

- Utilize o `zfs inherit -S` para reverter um valor de propriedade local para o valor recebido, se houver algum. Se uma propriedade não possuir 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 possuir 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`.
- É possível utilizar a opção `zfs receive -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    -
```

Em alguns casos, as propriedades do sistema de arquivos em um fluxo de envio podem não se aplicar ao sistema de arquivos de recebimento ou às propriedades do sistema de arquivos local, como o valor de propriedade `mountpoint`, que pode interferir em uma restauração.

Por exemplo, o sistema de arquivos `tank/data` tem a propriedade de compactação desativada. Um instantâneo do sistema de arquivos `tank/data` é enviado com propriedades (opção `p`) para um pool de backup e é recebido com a propriedade de compactação ativada.

```
# zfs get compression tank/data
NAME      PROPERTY  VALUE  SOURCE
tank/data compression off    default
# zfs snapshot tank/data@snap1
# zfs send -p tank/data@snap1 | zfs recv -o compression=on -d bpool
# zfs get -o all compression bpool/data
NAME      PROPERTY  VALUE  RECEIVED  SOURCE
bpool/data compression on      off      local
```

No exemplo, a propriedade `compression` é ativada quando o instantâneo é recebido em `bpool`. Portanto, para `bpool/data`, o valor de compactação é `on`.

Se este fluxo de instantâneos for enviado para um novo pool, `restorepool`, para fins de recuperação, você pode manter todas as propriedades originais do instantâneo. Nesse caso, você usaria o comando `zfs send -b` para restaurar as propriedades originais do instantâneo. Por exemplo:

```
# zfs send -b bpool/data@snap1 | zfs recv -d restorepool
# zfs get -o all compression restorepool/data
NAME                PROPERTY  VALUE  RECEIVED  SOURCE
restorepool/data    compression off      off        received
```

No exemplo, o valor de compactação é `off`, que representa o valor de compactação do instantâneo do sistema de arquivos `tank/data` original.

Se você tiver um valor de propriedade do sistema de arquivos local em um fluxo de instantâneos e quiser desativar a propriedade quando for recebida, use o comando `zfs receive -i`. Por exemplo, o comando a seguir envia um fluxo de instantâneos recursivo de sistemas de arquivos do diretório inicial com todas as propriedades do sistema de arquivos reservadas para um pool de backup, mas sem os valores de propriedade de cota:

```
# zfs send -R tank/home@snap1 | zfs recv -x quota bpool/home
# zfs get -r quota bpool/home
NAME                PROPERTY  VALUE  SOURCE
bpool/home           quota     none   local
bpool/home@snap1     quota     -      -
bpool/home/lori       quota     none   default
bpool/home/lori@snap1 quota     -      -
bpool/home/mark       quota     none   default
bpool/home/mark@snap1 quota     -      -
```

Se o instantâneo recursivo não foi recebido com a opção `-i`, a propriedade de cota seria definida nos sistemas de arquivos recebidos.

```
# zfs send -R tank/home@snap1 | zfs recv bpool/home
# zfs get -r quota bpool/home
NAME                PROPERTY  VALUE  SOURCE
bpool/home           quota     none   received
bpool/home@snap1     quota     -      -
bpool/home/lori       quota     10G    received
bpool/home/lori@snap1 quota     -      -
bpool/home/mark       quota     10G    received
bpool/home/mark@snap1 quota     -      -
```

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.
- Ao usar a opção `zfs send -r` sem a opção `-c` e ao usar a opção `zfs send -R`, os pacotes de fluxos omitem a origem de clones em algumas circunstâncias. Para obter mais informações, consulte [“Identificando fluxos de instantâneos do ZFS” na página 224](#).
- 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 serão 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` foi utilizado, apenas `snapD` (para todos os descendentes) será enviado.
- 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 6-1 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`.

EXEMPLO 6-1 Enviando e recebendo fluxos complexos de instantâneos do ZFS (Continuação)

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

EXEMPLO 6-1 Enviando e recebendo fluxos complexos de instantâneos do ZFS (Continuação)

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 sistema de arquivos `users` e seus descendentes, e enviar o fluxo replicado para outro pool, `users2`.

```
# zfs create users2 mirror c0t1d0 c1t1d0
# zfs receive -F -d users2 < /snaps/users-R
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
users	224K	33.2G	22K	/users
users@today	0	-	22K	-
users/user1	33K	33.2G	18K	/users/user1
users/user1@today	15K	-	18K	-
users/user2	18K	33.2G	18K	/users/user2
users/user2@today	0	-	18K	-
users/user3	18K	33.2G	18K	/users/user3
users/user3@today	0	-	18K	-
users2	188K	16.5G	22K	/users2
users2@today	0	-	22K	-
users2/user1	18K	16.5G	18K	/users2/user1
users2/user1@today	0	-	18K	-
users2/user2	18K	16.5G	18K	/users2/user2
users2/user2@today	0	-	18K	-
users2/user3	18K	16.5G	18K	/users2/user3
users2/user3@today	0	-	18K	-

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.

Uso de ACLs e atributos para proteger arquivos do Oracle Solaris ZFS

Este capítulo oferece informações sobre o uso das listas de controle de acesso (ACLs) para proteger os arquivos ZFS proporcionando permissões mais granulares que as permissões do UNIX padrão.

Este capítulo traz as seguintes seções:

- “Modelo de ACL do Solaris” na página 235
- “Definindo ACLs em arquivos ZFS” na página 242
- “Definindo e exibindo ACLs em arquivos ZFS no formato verboso” na página 245
- “Definindo e exibindo ACLs em arquivos ZFS no formato compacto” na página 255

Modelo de ACL do Solaris

As versões anteriores do Solaris 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 os seguintes:

- Baseado na especificação NFSv4 e semelhante às ACLs de estilo NT.
- Proporciona privilégios de acesso muito mais granulares. Para obter mais informações, consulte a [Tabela 7-2](#).
- É definido e exibido com os comandos `chmod` e `ls` em vez dos comandos `setfacl` e `getfacl`.

- Oferece semânticas de herança mais ricas para designar como os privilégios de acesso serão aplicados do diretório aos subdiretórios, e assim por diante. Para obter mais informações, consulte [“Herança da ACL” na página 240](#).

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 de esquema POSIX usam uma única entrada para definir que permissões serão aceitas e que permissões serão negadas. O novo modelo de ACL apresenta dois tipos de ACEs que afetam a verificação de acesso: ALLOW e DENY. Não é possível deduzir de nenhuma ACE que define um conjunto de permissões se as permissões definidas nesta ACE serão aceitas ou negadas.

A tradução entre ACLs de estilo NFSv4 e de esquema POSIX é a seguinte:

- Se usar utilitários com ACL, como os comandos `cp`, `mv`, `tar`, `cpio` ou `rcp`, para transferir arquivos UFS com ACLs para um sistema de arquivos ZFS, as ACLs de esquema POSIX são traduzidas para as ACLs de estilo NFSv4 equivalentes.
- Algumas ACLs de estilo NFSv4 são traduzidas para ACLs de esquema POSIX. Se uma ACL de estilo NFSv4 não for traduzida para uma ACL de esquema 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 `ufs restore` para restaurar os dados dentro de um sistema de arquivos ZFS. Se os dados originais possuírem ACLs de estilo POSIX, elas são convertidas em ACLs de estilo NFSv4.
- Se tentar definir uma ACL de estilo 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 esquema 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 224](#).

Descrições de sintaxe para definição de ACLs

São fornecidos dois formatos básicos de ACL:

- **ACL Trivial** – Contém apenas entradas tradicionais de usuário, grupo e proprietário do UNIX.
- **ACL Não Trivial** – Contém mais entradas que apenas proprietário, grupo e todos, ou inclui um conjunto de sinalizadores herdados ou as entradas são ordenadas de uma forma não tradicional.

Sintaxe para definição de ACLs comuns

```
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 obter uma descrição dos *tipos de entrada ACL*, consulte a [Tabela 7-1](#).

```
user or group:ACL-entry-ID=username or groupname
```

Identifica o *tipo de entrada ACL* da sintaxe de ACL explícita. O *tipo de entrada ACL* do usuário ou do grupo deve conter também o *ID da entrada ACL*, *nome do usuário* ou *nome do grupo*. Para obter uma descrição dos *tipos de entrada ACL*, consulte a [Tabela 7-1](#).

```
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 7-2](#).

```
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 7-4](#).

```
deny | allow
```

Identifica se as permissões de acesso são aceitas ou negadas.

No exemplo a seguir, nenhum valor *ACL-entry-ID* existe para *owner@*, *group@* ou *everyone@*..

```
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
```

O **2** ou a designação do *ID de índice* neste exemplo identifica a entrada ACL na ACL maior, que pode ter várias entradas para *owner*, *UIDs* específicos, *group* e *everyone*. Você pode especificar o *ID de índice* com o comando *chmod* para identificar que parte da ACL quer modificar. Você pode identificar, por exemplo, o ID de índice ID 3 como **A3** para o comando *chmod*, semelhante a:

```
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 7-1 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. Deve incluir o <i>ID da entrada ACL</i> , que contém um <i>nome de usuário</i> ou um <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. Deve incluir o <i>ID da entrada ACL</i> , que contém um <i>nome de grupo</i> ou um <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 7-2 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.
append_data	p	Ainda não implementado.
delete	d	Permissão para excluir um arquivo. Para obter mais informações sobre o comportamento da permissão específica delete, consulte a Tabela 7-3 .
delete_child	D	Permissão para excluir um arquivo ou diretório dentro de um diretório. Para obter mais informações sobre o comportamento da permissão específica delete_child, consulte a Tabela 7-3 .
execute	x	Permissão para executar um arquivo ou pesquisar o conteúdo de um diretório.
list_directory	r	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 da máscara de acesso, a entidade pode executar ls(1) e stat(2).
read_data	r	Permissão para ler o conteúdo do arquivo.
read_xattr	A	Permissão para ler os atributos estendidos de um arquivo ou realizar uma pesquisa no diretório dos atributos estendidos do arquivo.
synchronize	s	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	A	Permissão para alterar para um valor arbitrário os períodos de tempo associados a um arquivo ou diretório.
write_acl	C	Permissão para gravar a ACL ou habilidade para modificar a ACL com o comando chmod.

TABELA 7-2 Privilégios de acesso da ACL (Continuação)

Privilégio de Acesso	Privilégio de Acesso Compacto	Descrição
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 o grupo do qual o usuário é um membro. Se quiser alterar a propriedade do grupo ou arquivo para um usuário ou grupo arbitrário, o privilégio PRIV_FILE_CHOWN é exigido.

A tabela a seguir oferece detalhes adicionais sobre os comportamentos de ACL delete e delete_child.

TABELA 7-3 Comportamento de Permissão de ACL delete e delete_child

Permissões do Diretório Pai		Permissões do Objeto de Destino	
		A ACL permite a exclusão	A ACL nega a exclusão
			Permissão de exclusão não especificada
A ACL permite delete_child	Permitir	Permitir	Permitir
A ACL nega delete_child	Permitir	Negar	Negar
A ACL permite apenas write e execute	Permitir	Permitir	Permitir
A ACL nega write e execute	Permitir	Negar	Negar

Herança da ACL

O objetivo de usar a herança da ACL é que um arquivo ou diretório recém-criado possa herdar as ACLs que pretendem herdar, mas levando em consideração os bits de permissão 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 tabela abaixo.

TABELA 7-4 Sinalizadores de Herança da ACL

Sinalizador de Herança	Sinalizador de Herança Compacto	Descrição
<code>file_inherit</code>	<code>f</code>	Herda somente a ACL do diretório pai até os arquivos do diretório.
<code>dir_inherit</code>	<code>d</code>	Herda somente a ACL do diretório pai até os 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, você pode definir a diretriz de herança da ACL padrão no sistema de arquivos para que seja mais rigorosa ou menos rigorosa usando a propriedade `aclinherit` do sistema de arquivos. Para obter mais informações, consulte a próxima seção.

Propriedades da ACL

O sistema de arquivos do ZFS inclui as seguintes propriedades de ACL para determinar o comportamento específico da herança de ACL e interação de ACL com operações `chmod`.

- `aclinherit` – Determine o comportamento da herança de ACL. Os valores incluem o seguinte:
 - `discard` – Para novos objetos, nenhuma entrada ACL é herdada quando um arquivo ou diretório é criado. A ACL no arquivo ou diretório é igual ao modo de permissão 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 um modo determinado pelas ACEs herdáveis. Se não houver ACEs herdáveis que afetem o modo, então o modo será definido de acordo com o modo solicitado do aplicativo.

- `passthrough-x` – 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 que afeta o modo.

O modo padrão do `aclinherit` é `restricted`.

- `aclmode` – Modifica o comportamento de ACL quando um arquivo é criado inicialmente ou controla como uma ACL é modificada durante uma operação `chmod`. Os valores incluem:
 - `discard` – Um sistema de arquivos com uma propriedade `aclmode` de `discard` exclui todas as entradas de ACL que não representam o modo do arquivo. Este é o valor padrão.
 - `mask` – Um sistema de arquivos com uma propriedade `aclmode` de `mask` reduz as permissões de usuário ou grupo. As permissões são reduzidas, de forma que não são maiores que os bits de permissão do grupo, a menos que seja uma entrada de usuário com o mesmo UID do proprietário do arquivo ou diretório. Nesse caso, as permissões de ACL são reduzidas para que não fiquem maiores que os bits da permissão do proprietário. O valor da máscara também preserva a ACL nas alterações de modo, desde que uma operação explícita de definição da ACL não tenha sido executada.
 - `passthrough` – Um sistema de arquivos com uma propriedade `aclmode` de `passthrough` indica que não houve alterações na ACL além da geração das entradas de ACL necessárias para representar o novo modo do arquivo ou diretório.

O modo padrão de `aclmode` é `discard`.

Para obter mais informações sobre o uso da propriedade `aclmode`, consulte o [Exemplo 7-13](#).

Definindo ACLs em arquivos ZFS

Por serem implementadas com ZFS, as ACLs estão compostas de uma matriz de entradas ACLs. O ZFS oferece um modelo de ACL *puro*, no qual todos os arquivos têm uma ACL. Normalmente, a ACL é *comum* visto que representa somente as entradas `owner/group/other` tradicionais do UNIX.

Os arquivos ZFS ainda possuem bits de permissão e um modo, mas esses valores são mais de um cache do que a ACL representa. Se você alterar as permissões do arquivo, a ACL do arquivo será conseqüentemente atualizada. Além disso, se você remover uma ACL não-comum que permite que o usuário acesse o 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 ao grupo ou a 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 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 (negar) subsequente no mesmo conjunto de permissões da ACL.
- A permissão `write_acl` é concedida incondicionalmente ao proprietário, mesmo se a permissão for explicitamente negada. Caso contrário, todas as permissões não especificadas são negadas.

No caso de permissões negadas ou quando falta um acesso de permissão, o subsistema de privilégios determina que 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 usar os sinalizadores de herança da ACL. Para obter mais informações, consulte a [Tabela 7-4](#) e “[Definindo a herança da ACL em arquivos ZFS no formato verboso](#)” na [página 249](#).

Quando um novo arquivo é criado e dependendo do valor de `umask`, um padrão de ACL comum, semelhante ao ilustrado abaixo, é aplicado:

```
$ ls -lv file.1
-rw-r--r--  1 root    root      206663 Jun 23 15:06 file.1
 0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
    /read_attributes/write_attributes/read_acl/write_acl/write_owner
    /synchronize:allow
 1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
 2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
    :allow
```

Cada categoria de usuário (`owner@`, `group@`, `everyone@`) possui uma entrada de ACL neste exemplo.

Encontra-se abaixo uma descrição dessa ACL do arquivo:

0:owner@ O proprietário pode ler e modificar o conteúdo do arquivo (`read_data/write_data/append_data/read_xattr`). O proprietário também pode modificar os atributos do arquivo, tais como carimbos de data/hora, atributos estendidos e ACLs (`write_xattr/read_attributes/write_attributes/read_acl/write_acl`). Além disso, o proprietário pode modificar a propriedade do arquivo (`write_owner:allow`).

A permissão de acesso `synchronize` não está implementada atualmente.

1:group@ O grupo recebe permissões de leitura para o arquivo e os atributos do arquivo (`read_data/read_xattr/read_attributes/read_acl:allow`).

2:everyone@ Todos os que não são nem usuário nem grupo têm permissões de leitura para o arquivo e os atributos do arquivo (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, a ACL padrão do diretório é semelhante ao ilustrado abaixo:

```
$ ls -dv dir.1
drwxr-xr-x  2 root   root       2 Jul 20 13:44 dir.1
 0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/delete_child
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
 1:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
 2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Abaixo encontra-se uma descrição do diretório de ACL:

- 0:owner@ O proprietário pode ler e modificar o conteúdo do diretório (list_directory/read_data/add_file/write_data/add_subdirectory/append_data) e ler e modificar os atributos do arquivo, como registros de data/hora, atributos estendidos e ACLs (/read_xattr/write_xattr/read_attributes/write_attributes/read_acl/write_acl). Além disso, o proprietário pode pesquisar o conteúdo (execute), excluir um arquivo ou diretório (delete_child) e modificar a propriedade do diretório (write_owner:allow).
- A permissão de acesso synchronize não está implementada atualmente.
- 1:group@ O grupo pode listar e ler o conteúdo do diretório e os atributos do diretório. Além disso, o grupo tem permissão para executar pesquisas no conteúdo do diretório (list_directory/read_data/read_xattr/execute/read_attributes/read_acl/synchronize:allow).
- 2:everyone@ Todos os que não são nem usuário nem grupo têm permissão de leitura e execução para o conteúdo e os 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 237](#).

- 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 ACL
 - 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      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

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 255](#).

EXEMPLO 7-1 Modificando ACLs comuns em arquivos ZFS

Esta seção oferece exemplos de configuração e exibição de ACLs triviais, o que significa que apenas as entradas de UNIX, usuário e grupo tradicionais, entre outros, são incluídos na ACL.

No exemplo abaixo, existe uma ACL comum no `file.1`:

```
# ls -v file.1
-rw-r--r--  1 root    root      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

No exemplo abaixo, as permissões `write_data` são concedidas a `group@`.

```
# chmod A1=group@:read_data/write_data:allow file.1
# ls -v file.1
-rw-rw-r--  1 root    root      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/write_data:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

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      206695 Jul 20 13:43 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

EXEMPLO 7-2 Definindo ACLs não-comuns em arquivos ZFS

Esta seção oferece exemplos de definições e exibições de ACLs não-comuns.

No exemplo abaixo, 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 Jul 20 14:23 test.dir
0:user:gozer:list_directory/read_data/execute:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
```

EXEMPLO 7-2 Definindo ACLs não-comuns em arquivos ZFS (Continuação)

```

/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

No exemplo abaixo, as permissões `read_data/execute` são removidas do usuário `gozer`.

```

# chmod A0- test.dir
# ls -dv test.dir
drwxr-xr-x  2 root    root          2 Jul 20 14:23 test.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

EXEMPLO 7-3 A interação da ACL com permissões em arquivos ZFS

Os exemplos de ACL a seguir ilustram a interação entre definir ACLs e alterar os bits de permissão do arquivo ou diretório.

No exemplo abaixo, existe uma ACL comum em `file.2`:

```

# ls -v file.2
-rw-r--r--  1 root    root          2693 Jul 20 14:26 file.2
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow

```

No exemplo abaixo, as permissões `allow` da ACL são removidas de `everyone@`.

```

# chmod A2- file.2
# ls -v file.2
-rw-r-----  1 root    root          2693 Jul 20 14:26 file.2
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow

```

Nesta saída, os bits de permissão do arquivo são redefinidos de 644 para 640. As permissões de leitura de `everyone@` foram removidas com sucesso dos bits de permissão do arquivo quando as permissões `allow` da ACL foram removidas de `everyone@`.

EXEMPLO 7-3 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          2440 Jul 20 14:28 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, os bits de permissão são definidos 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          2440 Jul 20 14:28 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@`, o que representa 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          2440 Jul 20 14:28 file.3
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
2:group@:read_data/read_xattr/execute/read_attributes/read_acl
  /synchronize:allow
3:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
  /synchronize:allow
```

EXEMPLO 7-4 Restaurando ACLs comuns em arquivos ZFS

Você pode usar o comando `chmod` para remover todas as ACLs não-comuns em um arquivo ou diretório. (exemplo de)

No exemplo abaixo, existem duas ACEs não-comuns em `test5.dir`.

```
# ls -dv test5.dir
drwxr-xr-x+ 2 root      root          2 Jul 20 14:32 test5.dir
0:user:lp:read_data:file_inherit:deny
```


EXEMPLO 7-4 Restaurando ACLs comuns em arquivos ZFS (Continuação)

```

1:user:gozer:read_data:file_inherit:deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/delete_child
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
3:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
4:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

No exemplo abaixo, as ACLs não-comuns de usuários gozer e lp são removidas. A ACL restante contém os valores padrão para owner@, group@ e everyone@.

```

# chmod A- test5.dir
# ls -dv test5.dir
drwxr-xr-x  2 root    root          2 Jul 20 14:32 test5.dir
  0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
    /append_data/read_xattr/write_xattr/execute/delete_child
    /read_attributes/write_attributes/read_acl/write_acl/write_owner
    /synchronize:allow
  1:group@:list_directory/read_data/read_xattr/execute/read_attributes
    /read_acl/synchronize:allow
  2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
    /read_acl/synchronize:allow

```

Definindo a herança da ACL em arquivos ZFS no formato verboso

É possível determinar como as ACLs são herdadas ou não nos arquivos e diretórios. 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. Você deve especificar a herança de uma ACL em um arquivo ou diretório.

A propriedade `aclinherit` pode ser configurada globalmente em um sistema de arquivos. Por padrão, `aclinherit` está configurado como `restrito`.

Para obter mais informações, consulte [“Herança da ACL” na página 240](#).

EXEMPLO 7-5 Concedendo herança de ACL padrão

Por padrão, as ACLs não se propagam pela estrutura do diretório.

No exemplo abaixo, uma ACE não-comum de `read_data/write_data/execute` é aplicada no usuário gozer em `test.dir`.

```

# chmod A+user:gozer:read_data/write_data/execute:allow test.dir
# ls -dv test.dir

```

EXEMPLO 7-5 Concedendo herança de ACL padrão (Continuação)

```
drwxr-xr-x+ 2 root    root          2 Jul 20 14:53 test.dir
0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/delete_child
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

Se for criado um subdiretório `test.dir`, a ACE não se propaga para o usuário `gozer`. O usuário `gozer` teria acesso somente a `sub.dir` se as permissões em `sub.dir` lhe permitissem acessar como proprietário de arquivo, membro de grupo ou `everyone@`.

```
# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x 2 root    root          2 Jul 20 14:54 test.dir/sub.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/delete_child
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

EXEMPLO 7-6 Permitindo herança de ACL em arquivos e diretórios

Estas séries de exemplos identificam as ACEs do arquivo e do diretório aplicadas quando o sinalizador `file_inherit` está definido.

No exemplo a seguir, as permissões `read_data/write_data` são adicionadas para arquivos do diretório `test2.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 Jul 20 14:55 test2.dir
0:user:gozer:read_data/write_data:file_inherit:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/delete_child
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
```

EXEMPLO 7-6 Permitindo herança de ACL em arquivos e diretórios *(Continuação)*

No exemplo abaixo, as permissões do usuário gozer são aplicadas no arquivo `test2.dir/file.2` recém-criado. A herança de ACL permitida, `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 Jul 20 14:56 test2.dir/file.2
0:user:gozer:read_data:inherited:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

Pelo fato de `aclinherit` estar definida como modo padrão (`groupmask`) nesse sistema de arquivos, o usuário gozer não tem a permissão `write_data` no `file.2` porque a permissão de grupo do arquivo não lhe dá acesso.

Observe que a permissão `inherit_only`, aplicada quando os sinalizadores `file_inherit` ou `dir_inherit` estão definidos, é usada 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 Jun 23 15:21 test2.dir/subdir.2
0:user:gozer:list_directory/read_data/add_file/write_data:file_inherit
/inherit_only:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/read_attributes
/write_attributes/read_acl/write_acl/write_owner/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
```

As séries de exemplos abaixo identificam as ACLs do arquivo e do diretório aplicadas quando ambos sinalizadores, `file_inherit` e `dir_inherit`, estão definidos.

No exemplo abaixo, 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 Jul 20 15:00 test3.dir
```

EXEMPLO 7-6 Permitindo herança de ACL em arquivos e diretórios *(Continuação)*

```

0:user:gozer:list_directory/read_data/add_file/write_data/execute
:file_inherit/dir_inherit:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

# touch test3.dir/file.3
# ls -lv test3.dir/file.3
-rw-r--r--+ 1 root    root          0 Jun 23 15:25 test3.dir/file.3
0:user:gozer:read_data:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow

# mkdir test3.dir/subdir.1
# ls -dv test3.dir/subdir.1
drwxr-xr-x+ 2 root    root          2 Jun 23 15:26 test3.dir/subdir.1
0:user:gozer:list_directory/read_data/execute:file_inherit/dir_inherit
:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/read_attributes
/write_attributes/read_acl/write_acl/write_owner/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

Nesses exemplos, pelo fato de os bits de permissão do diretório pai em 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.

No exemplo abaixo, o usuário gozer tem permissões de leitura, gravação e execução herdadas por arquivos recém-criados, mas que não são propagadas para o conteúdo subsequente do diretório.

```

# chmod A+user:gozer:read_data/write_data/execute:file_inherit/no_propagate:allow
test4.dir
# ls -dv test4.dir
drwxr--r--+ 2 root    root          2 Mar  1 12:11 test4.dir
0:user:gozer:list_directory/read_data/add_file/write_data/execute
:file_inherit/no_propagate:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child

```

EXEMPLO 7-6 Permitindo herança de ACL em arquivos e diretórios (Continuação)

```

        /read_attributes/write_attributes/read_acl/write_acl/write_owner
        /synchronize:allow
2:group@:list_directory/read_data/read_xattr/read_attributes/read_acl
        /synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
        /synchronize:allow

```

Conforme ilustra o exemplo abaixo, as permissões de gozer de read_data/write_data/execute são reduzidas com base nas permissões do grupo responsável.

```

# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 Jun 23 15:28 test4.dir/file.4
0:user:gozer:read_data:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
        /read_attributes/write_attributes/read_acl/write_acl/write_owner
        /synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
        :allow

```

EXEMPLO 7-7 Herança da ACL com o modo de ACL herdada definido como passar

Se a propriedade aclinherit estiver definida como passthrough no sistema de arquivos tank/cindy, o usuário gozer pode herdar a ACL aplicada em test4.dir para o file.4 recém-criado da seguinte forma:

```

# zfs set aclinherit=passthrough tank/cindy
# touch test4.dir/file.4
# ls -v test4.dir/file.4
-rw-r--r--+ 1 root    root          0 Jun 23 15:35 test4.dir/file.4
0:user:gozer:read_data:allow
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
        /read_attributes/write_attributes/read_acl/write_acl/write_owner
        /synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
        :allow

```

EXEMPLO 7-8 Herança da ACL com o Modo de Herança de ACL Definido como Descartar

Se a propriedade aclinherit estiver definida como discard em um sistema de arquivos, as ACLs podem ser potencialmente descartadas quando os bits de permissão de um diretório se alteram. Por exemplo:

```

# zfs set aclinherit=discard tank/cindy
# chmod A=user:gozer:read_data/write_data/execute:dir_inherit:allow test5.dir
# ls -dv test5.dir
drwxr-xr-x+ 2 root    root          2 Jul 20 14:18 test5.dir
0:user:gozer:list_directory/read_data/add_file/write_data/execute

```

EXEMPLO 7-8 Herança da ACL com o Modo de Herança de ACL Definido como Descartar
(Continuação)

```

:dir_inherit:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

Se, mais tarde, você decidir comprimir os bits de permissão 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 Jul 20 14:18 test5.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/read_attributes/read_acl
/synchronize:allow

```

EXEMPLO 7-9 Herança da ACL com o modo de herança da ACL definido como não permitir

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 ilustra também como você pode especificar duas ACEs no mesmo comando `chmod`.

```

# zfs set aclinherit=noallow tank/cindy
# chmod A+user:gozer:read_data:file_inherit:deny,user:lp:read_data:file_inherit:allow
test6.dir
# ls -dv test6.dir
drwxr-xr-x+  2 root    root          2 Jul 20 14:22 test6.dir
0:user:gozer:read_data:file_inherit:deny
1:user:lp:read_data:file_inherit:allow
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
/append_data/read_xattr/write_xattr/execute/delete_child
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
3:group@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow
4:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

Conforme mostra o exemplo abaixo, quando um novo arquivo é criado, a ACL que concede a permissão `read_data` é descartada.

EXEMPLO 7-9 Herança da ACL com o modo de herança da ACL definido como não permitir
(Continuação)

```
# touch test6.dir/file.6
# ls -v test6.dir/file.6
-rw-r--r--+ 1 root    root          0 Jun 15 12:19 test6.dir/file.6
0:user:gozer:read_data:inherited:deny
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

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 permissões compactas são listadas na [Tabela 7-2](#) e na [Tabela 7-4](#).

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 23 15:06 file.1
owner@:rw-p--aARWcCos:-----:allow
group@:r-----a-R-c--s:-----:allow
everyone@:r-----a-R-c--s:-----:allow
```

A saída de ACL compacta é descrita da seguinte forma:

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/hora, atributos estendidos e ACLs (a=read_attributes, W=write_xattr, R= read_xattr, A=write_attributes, c=read_acl, C= write_acl). Além disso, o proprietário pode modificar a propriedade do arquivo (o=write_owner).

A permissão de acesso synchronize (s) não está implementada atualmente.

group@ O grupo recebe permissões de leitura para o arquivo (r= read_data) e os atributos do arquivo (a=read_attributes, R=read_xattr, c= read_acl).

A permissão de acesso synchronize (s) não está implementada atualmente.

everyone@ Todos os que não são nem usuários nem grupos têm permissões de leitura para o arquivo e para os atributos do arquivo (r=read_data, a=append_data, R=read_xattr, c=read_acl e s= synchronize).

A permissão de acesso synchronize (s) 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`.
- Os hífen (-), que indicam "sem permissões", podem ser removidos e somente as letras requeridas devem 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 245](#).

EXEMPLO 7-10 Definindo e exibindo ACLs em formato compacto

No exemplo abaixo, existe uma ACL comum no `file.1`:

```
# ls -V file.1
-rw-r--r-- 1 root    root      206663 Jun 23 15:06 file.1
      owner@:rw-p--aARWcCos:-----:allow
      group@:r-----a-R-c--s:-----:allow
      everyone@:r-----a-R-c--s:-----:allow
```

Neste exemplo, as permissões read_data/execute são adicionadas ao usuário gozer em `file.1`.

```
# chmod A+user:gozer:rx:allow file.1
# ls -V file.1
-rw-r--r--+ 1 root    root      206663 Jun 23 15:06 file.1
      user:gozer:r-x-----:-----:allow
      owner@:rw-p--aARWcCos:-----:allow
      group@:r-----a-R-c--s:-----:allow
      everyone@:r-----a-R-c--s:-----:allow
```

No exemplo abaixo, o usuário gozer tem permissões de leitura, gravação e execução que são herdadas pelos novos arquivos e diretórios através do formato compacto de ACL.

```
# chmod A+user:gozer:rwX:fd:allow dir.2
# ls -dV dir.2
drwxr-xr-x+ 2 root    root      2 Jun 23 16:04 dir.2
      user:gozer:rwX-----:fd----:allow
      owner@:rwxp--aARWcCos:-----:allow
      group@:r-x---a-R-c--s:-----:allow
      everyone@:r-x---a-R-c--s:-----:allow
```


EXEMPLO 7-10 Definindo e exibindo ACLs em formato compacto (Continuação)

Você também pode recortar e colar os sinalizadores de permissões e de herança da saída `ls -lV` no formato compacto `chmod`. Por exemplo, para duplicar os sinalizadores de permissões e de herança no `dir.2` do usuário `gozer` para o usuário `cindy` em `dir.2`, copie e cole os sinalizadores de permissão e de herança (`rwX-----:fd---:allow`) no comando `chmod`. Por exemplo:

```
# chmod A+user:cindy:rwX-----:fd---:allow dir.2
# ls -lV dir.2
drwxr-xr-x+ 2 root    root          2 Jun 23 16:04 dir.2
      user:cindy:rwX-----:fd---:allow
      user:gozer:rwX-----:fd---:allow
      owner@:rwxp--aARWcCos:-----:allow
      group@:r-x---a-R-c-s:-----:allow
      everyone@:r-x---a-R-c-s:-----:allow
```

EXEMPLO 7-11 Herança da ACL com o modo de ACL herdada definido como passar

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 o valor da propriedade estiver definido como `passthrough`, os arquivos são criados com um modo de permissão determinado pelas ACEs herdáveis. Se não houver ACEs herdáveis que afetem o modo de permissão, então o modo de permissão será definido de acordo com o modo solicitado do aplicativo.

Os exemplos seguintes usam a sintaxe compacta de ACL para mostrar como herdar bits de permissão definindo o modo `aclinherit` como `passthrough`.

Neste exemplo, uma ACL está definida no `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@`.

```
# zfs set aclinherit=passthrough tank/cindy
# pwd
/tank/cindy
# mkdir test1.dir

# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow
test1.dir
# ls -lV test1.dir
drwxrwx---+ 2 root    root          2 Jun 23 16:10 test1.dir
      owner@:rwxpdDaARWcCos:fd---:allow
      group@:rwxp-----:fd---:allow
      everyone@:-----:fd---:allow
```

Neste exemplo, o arquivo recém-criado herda a ACL que foi especificada para ser herdada nos arquivos recém-criados.

EXEMPLO 7-11 Herança da ACL com o modo de ACL herdada definido como passar (Continuação)

```
# cd test1.dir
# touch file.1
# ls -V file.1
-rwxrwx---+ 1 root    root          0 Jun 23 16:11 file.1
      owner@: rwxpdDaARwCcos:-----:allow
      group@: rwxp-----:-----:allow
      everyone@:-----:-----:allow
```

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 23 16:13 subdir.1
      owner@: rwxpdDaARwCcos:fd----:allow
      group@: rwxp-----:fd----:allow
      everyone@:-----:fd----:allow
```

As entradas fd---- 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/cindy
# mkdir test2.dir
# cd test2.dir
# touch file.2
# ls -V file.2
-rw-r--r-- 1 root    root          0 Jun 23 16:15 file.2
      owner@: rw-p--aARwCcos:-----:allow
      group@: r-----a-R-c--s:-----:allow
      everyone@: r-----a-R-c--s:-----:allow
```

EXEMPLO 7-12 Herança da ACL com o modo de ACL herdada definido como passar X

Quando `aclinherit=passthrough-x` estiver habilitado, os arquivos são criados com a permissão de executar (x) para o `owner@`, `group@` ou `everyone@`, mas somente de 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 o modo `aclinherit` como `passthrough-x`.

```
# zfs set aclinherit=passthrough-x tank/cindy
```

A seguinte ACL é definida em `/tank/cindy/test1.dir` para fornecer uma herança de ACL executável para arquivos para `owner@`.

```
# chmod A=owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow test1.dir
# ls -Vd test1.dir
```

EXEMPLO 7-12 Herança da ACL com o modo de ACL herdada definido como passar X (Continuação)

```
drwxrwx---+ 2 root    root          2 Jun 23 16:17 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 -lV test1.dir/file1
-rw-rw---+ 1 root    root          0 Jun 23 16:18 test1.dir/file1
  owner@: rw-pdDaARWcCos:-----:allow
  group@: rw-p-----:-----:allow
  everyone@:-----:-----:allow
```

A seguir, um executável denominado t é gerado com o uso do compilador cc no diretório testdir.

```
# cc -o t t.c
# ls -lV t
-rwxrwx---+ 1 root    root          7396 Dec  3 15:19 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@.

EXEMPLO 7-13 Interação de ACL com Operações chmod em Arquivos ZFS

Os exemplos a seguir ilustram como os valores de propriedade aclmode e aclinherit influenciam a interação de ACLs existentes com uma operação chmod que altera as permissões de arquivo ou diretório para reduzir ou expandir quaisquer permissões de ACL existentes para ser consistente com o grupo de propriedade.

Neste exemplo, a propriedade aclmode é definida como mask e a propriedade aclinherit é definida como restricted. As permissões de ACL deste exemplo são exibidas em modo compacto, o que ilustra mais facilmente as permissões de alteração.

O arquivo original e as permissões de ACL e propriedade são os seguintes:

```
# zfs set aclmode=mask pond/whoville
# zfs set aclinherit=restricted pond/whoville

# ls -lV file.1
-rwxrwx---+ 1 root    root          206695 Aug 30 16:03 file.1
  user:amy:r-----a-R-c---:-----:allow
  user:rory:r-----a-R-c---:-----:allow
  group:sysadmin:rw-p--aARWc---:-----:allow
```

EXEMPLO 7-13 Interação de ACL com Operações chmod em Arquivos ZFS (Continuação)

```

group:staff:rw-p--aARWc---:-----:allow
owner@:rwxp--aARWcCos:-----:allow
group@:rwxp--aARWc--s:-----:allow
everyone@:-----a-R-c--s:-----:allow

```

Uma operação chown altera a propriedade do arquivo no `file.1` e a saída é vista pelo usuário proprietário, amy. Por exemplo:

```

# chown amy:staff file.1
# su - amy
$ ls -lv file.1
-rwxrwx---+ 1 amy      staff      206695 Aug 30 16:03 file.1
      user:amy:r-----a-R-c---:-----:allow
      user:ror:r-----a-R-c---:-----:allow
group:sysadmin:rw-p--aARWc---:-----:allow
group:staff:rw-p--aARWc---:-----:allow
owner@:rwxp--aARWcCos:-----:allow
group@:rwxp--aARWc--s:-----:allow
everyone@:-----a-R-c--s:-----:allow

```

A operação chmod a seguir altera as permissões para um modo mais restritivo. Neste exemplo, o grupo `sysadmin` modificado e as permissões de ACL do grupo `staff` não excedem as permissões do grupo proprietário.

```

$ chmod 640 file.1
$ ls -lv file.1
-rw-r-----+ 1 amy      staff      206695 Aug 30 16:03 file.1
      user:amy:r-----a-R-c---:-----:allow
      user:ror:r-----a-R-c---:-----:allow
group:sysadmin:r-----a-R-c---:-----:allow
group:staff:r-----a-R-c---:-----:allow
owner@:rw-p--aARWcCos:-----:allow
group@:r-----a-R-c--s:-----:allow
everyone@:-----a-R-c--s:-----:allow

```

A operação chmod a seguir altera as permissões para um modo menos restritivo. Neste exemplo, as permissões de ACL do grupo `sysadmin` e do grupo `staff` são restauradas para permitir as mesmas permissões que as do grupo proprietário.

```

$ chmod 770 file.1
$ ls -lv file.1
-rwxrwx---+ 1 amy      staff      206695 Aug 30 16:03 file.1
      user:amy:r-----a-R-c---:-----:allow
      user:ror:r-----a-R-c---:-----:allow
group:sysadmin:rw-p--aARWc---:-----:allow
group:staff:rw-p--aARWc---:-----:allow
owner@:rwxp--aARWcCos:-----:allow
group@:rwxp--aARWc--s:-----:allow
everyone@:-----a-R-c--s:-----:allow

```

Administração delegada do ZFS do Oracle Solaris

Este capítulo descreve como utilizar a administração delegada para permitir que usuários não-privilegiados efetuem tarefas de administração no ZFS.

Este capítulo traz as seguintes seções:

- “Visão geral da administração delegada de ZFS” na página 261
- “Delegando permissões do ZFS” na página 262
- “Exibindo permissões delegadas do ZFS (Exemplos)” na página 270
- “Delegando permissões do ZFS (Exemplos)” na página 266
- “Removendo permissões delegadas ZFS (exemplos)” na página 272

Visão geral da administração delegada de ZFS

A administração delegada ZFS permite distribuir permissões refinadas a usuários específicos, grupos ou a todos. São suportados dois tipos de permissões delegadas:

- Permissões individuais podem ser delegadas explicitamente como `create`, `destroy`, `mount`, `snapshot`, a assim por diante.
- 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 símbolo `@`, os caracteres restantes no nome do conjunto apresentam as mesmas restrições que os nomes de um sistema de arquivos ZFS normal.

A administração delegada ZFS fornece recursos semelhantes ao modelo de segurança RBAC. O modelo de delegação do ZFS oferece as seguintes vantagens na administração de pools de armazenamento e sistemas de arquivos ZFS:

- As permissões seguem o pool de armazenamento do ZFS sempre que um 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.
- É possível delegar permissões para sistemas de arquivos específicos. 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 informações sobre a utilização do RBAC para gerenciar tarefas de administração gerais do Oracle 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 pool. 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

É possível utilizar o comando `zfs allow` para delegar permissões nos sistemas de arquivos do ZFS a usuários não raiz das seguintes formas:

- As permissões individuais podem ser delegadas a um usuário, grupo ou a todos.
- Os grupos de permissões individuais podem ser delegados como um *conjunto de permissões* a um usuário, grupo ou a todos.
- É possível delegar permissões localmente somente para o sistema de arquivos atual ou para todos os descendentes do sistema de arquivos 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 permissão para conceder permissões que possui a outro usuário.	Também deve ter a permissão que está sendo concedida.
<code>clone</code>	A permissão para clonar quaisquer instantâneos do conjunto de dados.	Também deve possuir as permissões <code>create</code> e <code>mount</code> no sistema de arquivos original.
<code>create</code>	A permissão para criar conjuntos de dados descendentes.	Também deve possuir a permissão <code>mount</code> .
<code>destroy</code>	A permissão para destruir um conjunto de dados.	Também deve possuir a permissão <code>mount</code> .
<code>diff</code>	A permissão para identificar caminhos em um conjunto de dados.	Usuários não-raiz precisam desta permissão para usar o comando <code>zfs diff</code> .
<code>hold</code>	A permissão para manter um instantâneo.	
<code>mount</code>	A permissão para montar e desmontar um sistema de arquivos para criar e destruir os links do dispositivo de volume.	
<code>promote</code>	A permissão para promover um clone para um conjunto de dados.	Também deve possuir as permissões <code>mount</code> e <code>promover</code> no sistema de arquivos original.
<code>receive</code>	A permissão para criar sistemas de arquivos descendentes com o comando <code>zfs receive</code> .	Também deve possuir as permissões <code>mount</code> e <code>create</code> .
<code>release</code>	A permissão para liberar um instantâneo mantido, o que pode destruir o instantâneo.	
<code>rename</code>	A permissão para renomear um conjunto de dados.	Também deve possuir as permissões <code>create</code> e <code>mount</code> no novo pai.
<code>reversão</code>	A permissão para reverter um instantâneo.	
<code>send</code>	A permissão para enviar um fluxo de instantâneo.	
<code>share</code>	A permissão para compartilhar e descompartilhar um sistema de arquivos.	Deve ter <code>share</code> e <code>sharenfs</code> para criar um compartilhamento de NFS. Deve ter <code>share</code> e <code>sharenfs</code> para criar um compartilhamento de SMB.

Permissão (Subcomando)	Descrição	Dependências
instantâneo	A permissão para criar um instantâneo de um conjunto de dados.	

É possível delegar os conjuntos de permissões a seguir, mas a permissão pode estar limitada para a permissão de acesso, leitura ou alteração:

- groupquota
- groupused
- userprop
- userquota
- userused

Além disso, é possível delegar as seguintes administrações das propriedades ZFS aos usuários não raiz:

- aclinherit
- aclmode
- atime
- canmount
- casesensitivity
- soma de verificação
- compression
- copies
- devices
- exec
- logbias
- mountpoint
- nbmand
- normalization
- primarycache
- quota
- readonly
- recordsize
- refquota
- refreservation
- reservation
- rstchown
- secondarycache
- setuid
- sharenfs
- sharesmb
- snapdir
- sync
- utf8only

- version
- volblocksize
- volsize
- vscan
- xattr
- zoned

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 179](#).

Delegação de permissões do ZFS (**zfs allow**)

A sintaxe **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** delega permissões do tempo de 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 identificação dos sistemas de arquivos ZFS, mas o nome deve começar com o sinal de arroba (@) e não pode ter mais de 64 caracteres em comprimento.

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 sistema de arquivos 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 sistemas de arquivos descendentes e não a este sistema de arquivos, a menos que a opção `-l` também seja especificada. Se nenhuma das opções forem especificadas, então as permissões são concedidas para o sistema de arquivos ou volume e a todos os seus descendentes.

Removendo as permissões delegadas do ZFS (`zfs unallow`)

É possível remover as permissões delegadas anteriormente com o comando `zfs unallow`.

Por exemplo, suponha que você delegou as permissões `create`, `destroy`, `mount` e `snapshot` como a seguir:

```
# zfs allow cindy create,destroy,mount,snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy -----
Local+Descendent permissions:
    user cindy create,destroy,mount,snapshot
```

Para remover estas permissões, é necessário utilizar a sintaxe a seguir:

```
# zfs unallow cindy tank/home/cindy
# zfs allow tank/home/cindy
```

Delegando permissões do ZFS (Exemplos)

EXEMPLO 8-1 Delegando permissões a um usuário individual

Ao delegar as permissões `create` e `mount` a um usuário individual, é necessário ter certeza de que o usuário tem as permissões no ponto de montagem subjacente.

Por exemplo, para delegar ao usuário as permissões `mark create` e `mount` no sistema de arquivos `tank`, defina primeiro as permissões:

```
# chmod A+user:mark:add_subdirectory:fd:allow /tank/home
```

A seguir, utilize o comando `zfs allow` para delegar as permissões `create`, `destroy` e `mount`. Por exemplo:

```
# zfs allow mark create,destroy,mount tank/home
```

Agora, o usuário `mark` pode criar seus próprios sistemas de arquivos no sistema de arquivos `tank/home`. Por exemplo:

EXEMPLO 8-1 Delegando permissões a um usuário individual *(Continuação)*

```
# su mark
mark$ zfs create tank/home/mark
mark$ ^D
# su lp
$ zfs create tank/home/lp
cannot create 'tank/home/lp': permission denied
```

EXEMPLO 8-2 Delegando permissões create e destroy a um grupo

O exemplo seguinte 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/home`, bem como destruir seus próprios sistemas de arquivos. No entanto, os membros do grupo `staff` não podem destruir os sistemas de arquivos de outros.

```
# zfs allow staff create,mount tank/home
# zfs allow -c create,destroy tank/home
# zfs allow tank/home
---- Permissions on tank/home -----
Create time permissions:
    create,destroy
Local+Descendent permissions:
    group staff create,mount
# su cindy
cindy% zfs create tank/home/cindy/files
cindy% exit
# su mark
mark% zfs create tank/home/mark/data
mark% exit
cindy% zfs destroy tank/home/mark/data
cannot destroy 'tank/home/mark/data': permission denied
```

EXEMPLO 8-3 Delegando permissões no nível correto do sistema de arquivos

Certifique-se de ter delegado permissão aos usuários no nível correto do sistema de arquivos. Por exemplo, ao usuário `mark` são delegadas as permissões `create`, `destroy` e `mount` para os sistemas de arquivos locais e descendentes. Ao usuário `mark`, é concedida a permissão local para realizar instantâneos do sistema de arquivos `tank/home`, mas não a permissão para realizar instantâneos do seu próprio sistema de arquivos. Assim, ele não tem a permissão `snapshot` delegada no nível correto do sistema de arquivos.

```
# zfs allow -l mark snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home -----
Create time permissions:
    create,destroy
Local permissions:
    user mark snapshot
Local+Descendent permissions:
    group staff create,mount
# su mark
mark$ zfs snapshot tank/home@snap1
```

EXEMPLO 8-3 Delegando permissões no nível correto do sistema de arquivos *(Continuação)*

```
mark$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

Para conceder ao usuário mark a permissão no nível do descendente, utilize a opção `zfs allow -d`. Por exemplo:

```
# zfs unallow -l mark snapshot tank/home
# zfs allow -d mark snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home -----
Create time permissions:
    create,destroy
Descendent permissions:
    user mark snapshot
Local+Descendent permissions:
    group staff create,mount
# su mark
$ zfs snapshot tank/home@snap2
cannot create snapshot 'tank/home@snap2': permission denied
$ zfs snapshot tank/home/mark@snappy
```

Agora, o usuário mark pode apenas criar um instantâneo abaixo do nível do sistema de arquivos tank/home.

EXEMPLO 8-4 Definindo e usando permissões delegadas complexas

É possível delegar permissões específicas a usuários ou grupos. Por exemplo, o comando `zfs allow` a seguir delega permissões específicas ao grupo `staff`. Além disso, as permissões `destroy` e `snapshot` são delegadas após a criação dos sistemas de arquivos tank/home.

```
# zfs allow staff create,mount tank/home
# zfs allow -c destroy,snapshot tank/home
# zfs allow tank/home
---- Permissions on tank/home -----
Create time permissions:
    create,destroy,snapshot
Local+Descendent permissions:
    group staff create,mount
```

Como o usuário mark é membro do grupo `staff`, ele pode criar sistemas de arquivos tank/home. Além disso, o usuário mark pode criar um instantâneo de tank/home/mark2 porque possui as permissões específicas para isso. Por exemplo:

```
# su mark
$ zfs create tank/home/mark2
$ zfs allow tank/home/mark2
---- Permissions on tank/home/mark2 -----
Local permissions:
    user mark create,destroy,snapshot
---- Permissions on tank/home -----
Create time permissions:
    create,destroy,snapshot
```

EXEMPLO 8-4 Definindo e usando permissões delegadas complexas (Continuação)

Local+Descendent permissions:
group staff create,mount

Mas o usuário mark não pode criar um instantâneo em tank/home/mark porque ele não possui a permissão específica para fazê-lo. Por exemplo:

```
$ zfs snapshot tank/home/mark@snap1
cannot create snapshot 'tank/home/mark@snap1': permission denied
```

Neste exemplo, usuário mark possui permissão create no diretório home, o que significa que ele pode criar instantâneos. Esta situação é útil quando o sistema de arquivos está montado em NFS.

```
$ cd /tank/home/mark2
$ ls
$ cd .zfs
$ ls
shares snapshot
$ cd snapshot
$ ls -l
total 3
drwxr-xr-x  2 mark  staff          2 Sep 27 15:55 snap1
$ pwd
/tank/home/mark2/.zfs/snapshot
$ mkdir snap2
$ zfs list
# zfs list -r tank/home
NAME                                USED  AVAIL  REFER  MOUNTPOINT
tank/home/mark                     63K   62.3G   32K    /tank/home/mark
tank/home/mark2                     49K   62.3G   31K    /tank/home/mark2
tank/home/mark2@snap1              18K         -   31K    -
tank/home/mark2@snap2               0         -   31K    -
$ ls
snap1 snap2
$ rmdir snap2
$ ls
snap1
```

EXEMPLO 8-5 Definindo e usando um conjunto de permissões delegadas do ZFS

O exemplo a seguir mostra como criar um conjunto de permissões @myset e delegar o conjunto de permissões ao grupo staff para o sistema de arquivos tank. O usuário cindy, membro do grupo staff, possui a permissão para criar um sistema de arquivos em tank. Entretanto, o usuário lp não possui permissão para criar um sistema de arquivos em tank.

```
# zfs allow -s @myset create,destroy,mount,snapshot,promote,clone,readonly tank
# zfs allow tank
---- Permissions on tank -----
Permission sets:
      @myset clone,create,destroy,mount,promote,readonly,snapshot
# zfs allow staff @myset,rename tank
# zfs allow tank
```

EXEMPLO 8-5 Definindo e usando um conjunto de permissões delegadas do ZFS (Continuação)

```

---- Permissions on tank -----
Permission sets:
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions:
    group staff @myset,rename
# chmod A+group:staff:add_subdirectory:fd:allow tank
# su cindy
cindy% zfs create tank/data
cindy% zfs allow tank
---- Permissions on tank -----
Permission sets:
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Local+Descendent permissions:
    group staff @myset,rename
cindy% ls -l /tank
total 15
drwxr-xr-x  2 cindy  staff      2 Jun 24 10:55 data
cindy% exit
# su lp
$ zfs create tank/lp
cannot create 'tank/lp': permission denied

```

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 no conjunto de dados específico. A saída contém os seguintes componentes:

- Conjuntos de permissões
- Permissões individuais ou permissões do tempo de criação
- Conjunto de dados local
- Conjuntos de dados locais e descendentes
- Conjuntos de dados descendentes somente

EXEMPLO 8-6 Exibindo permissões básicas de administração delegada

A saída a seguir indica que o usuário cindy possui permissões de instantâneo create, destroy, mount no sistema de arquivos tank/cindy.

```

# zfs allow tank/cindy
-----
Local+Descendent permissions on (tank/cindy)
    user cindy create,destroy,mount,snapshot

```

EXEMPLO 8-7 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 depois da criação do sistema de arquivos, o conjunto de permissões `@eng` e a permissão para definir a propriedade `mountpoint` são delegados.
- Ao usuário `tom`, é delegado 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 delegado 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 delegado o conjunto de permissões `@basic` apenas 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
---- Permissions on pool/fred -----
Permission sets:
    @eng create,destroy,snapshot,mount,clone,promote,rename
    @simple create,mount
Create time permissions:
    @eng,mountpoint
Local permissions:
    user tom @eng
    user joe create,destroy,mount
Local+Descendent permissions:
    user fred @basic,share,rename
    user barney @basic
    group staff @basic
---- Permissions on pool -----
Permission sets:
    @simple create,destroy,mount
Local permissions:
    group staff @simple
```

Removendo permissões delegadas ZFS (exemplos)

É possível utilizar o comando `zfs unallow` para remover as permissões delegadas. Por exemplo, o usuário `cindy` possui permissões `create`, `destroy`, `mount` e `snapshot` no sistema de arquivos `tank/cindy`.

```
# zfs allow cindy create,destroy,mount,snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy -----
Local+Descendent permissions:
    user cindy create,destroy,mount,snapshot
```

A sintaxe seguinte de `zfs unallow` remove a permissão de criação de instantâneo do usuário `cindy` do sistema de arquivos `tank/home/cindy`:

```
# zfs unallow cindy snapshot tank/home/cindy
# zfs allow tank/home/cindy
---- Permissions on tank/home/cindy -----
Local+Descendent permissions:
    user cindy create,destroy,mount
cindy% zfs create tank/home/cindy/data
cindy% zfs snapshot tank/home/cindy@today
cannot create snapshot 'tank/home/cindy@today': permission denied
```

Como outro exemplo, o usuário `mark` possui as seguintes permissões no sistema de arquivos `tank/home/mark`:

```
# zfs allow tank/home/mark
---- Permissions on tank/home/mark -----
Local+Descendent permissions:
    user mark create,destroy,mount
-----
```

A sintaxe `zfs unallow` a seguir remove todas as permissões para o usuário `mark` do sistema de arquivos `tank/home/mark`:

```
# zfs unallow mark tank/home/mark
```

A sintaxe seguinte de `zfs unallow` remove um conjunto de permissões do sistema de arquivos `tank`.

```
# zfs allow tank
---- Permissions on tank -----
Permission sets:
    @myset clone,create,destroy,mount,promote,readonly,snapshot
Create time permissions:
    create,destroy,mount
Local+Descendent permissions:
    group staff create,mount
# zfs unallow -s @myset tank
# zfs allow tank
---- Permissions on tank -----
Create time permissions:
```



```
create,destroy,mount  
Local+Descendent permissions:  
group staff create,mount
```


Tópicos avançados do Oracle Solaris ZFS

Este capítulo descreve os volumes ZFS, o uso do ZFS em um sistema Solaris com zonas instaladas, pools de raiz alternativa e perfis de direitos do ZFS.

Este capítulo contém as seguintes seções:

- “Volumes ZFS” na página 275
- “Usando ZFS em um sistema Solaris com zonas instaladas” na página 278
- “Usando pools de raiz alternativa do ZFS” na página 284

Volumes ZFS

Um volume ZFS consiste em um conjunto de dados que representa um dispositivo de bloco. Os volumes ZFS são identificados como dispositivos no diretório `/dev/zvol/{dsk, rdsk}/pool`.

No seguinte exemplo, 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 5–1](#).

Você pode exibir as informações de propriedade do volume ZFS usando o comando `zfs get` ou `zfs get all`. Por exemplo:

```
# zfs get all tank/vol
```

Um ponto de interrogação (?) exibido para `volsize` na saída `zfs get` indica um valor desconhecido pela ocorrência de um erro de E/S. Por exemplo:

```
# zfs get -H volsize tank/vol
tank/vol          volsize ?          local
```

Um erro de E/S geralmente indica um problema com um dispositivo do pool. Para obter informações sobre problemas no dispositivo do pool, consulte [“Identificando Problemas com Pools de Armazenamento do ZFS”](#) na página 290.

Se estiver usando um sistema Solaris com zonas instaladas, você não poderá criar ou clonar um volume ZFS em uma zona não global. Qualquer tentativa para tal irá falhar. Para obter informações sobre o uso de volumes ZFS em uma zona global, consulte [“Adicionando volumes ZFS a uma zona não global”](#) na página 281.

Usando um volume ZFS como um dispositivo de permuta ou despejo

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 pool 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 pool 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/
Savecore enabled: yes
```

Se você precisar alterar a área de intercâmbio ou o dispositivo de dump após a instalação 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. Em seguida, adicione uma entrada para o dispositivo de permuta no arquivo `/etc/vfstab`. 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 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 156.](#)

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 12, “Configuring Oracle Solaris iSCSI Targets \(Tasks\)”](#), 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 zonas instaladas

As seguintes seções descrevem como utilizar o ZFS em um sistema Oracle Solaris com zonas:

- [“Adicionando sistemas de arquivos ZFS a uma zona não global” na página 279](#)
- [“Delegando conjuntos de dados a uma zona não global” na página 280](#)
- [“Adicionando volumes ZFS a uma zona não global” na página 281](#)
- [“Usando pools de armazenamento do ZFS dentro de uma zona” na página 282](#)
- [“Gerenciando propriedades do ZFS dentro de uma zona” na página 282](#)
- [“Entendendo a propriedade zoned” na página 283](#)

Para obter informações sobre a configuração de zonas em um sistema com um sistema de arquivos raiz do ZFS que será migrado ou terá patches instalados com o Live Upgrade, consulte [“Usando o Live Upgrade para migrar ou atualizar um sistema com zonas \(Solaris 10 10/08\)” na página 139](#) ou [“Utilizando o Oracle Solaris Live Upgrade para migrar ou atualizar um sistema com zonas \(no mínimo Solaris 10 5/09\)” na página 144](#).

Leve em consideração os seguintes pontos ao associar os conjuntos de dados ZFS a zonas:

- Você pode adicionar um sistema de arquivos ou clone do ZFS a uma zona não global com ou sem delegar o controle administrativo.
- É possível adicionar um volume do ZFS como um dispositivo em zonas não globais.
- No momento, não é possível associar instantâneos do ZFS a zonas.

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 zona não global compartilhe espaço em disco com a zona global, embora o administrador de zonas não possa controlar as propriedades ou criar novos sistemas de arquivos na hierarquia do sistema de arquivos subjacente. Esta operação é idêntica a adicionar outro tipo de sistema de arquivos a uma zona, 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 zona não global, proporcionando controle total do administrador de zonas sobre o conjunto de dados e seus filhos. O administrador de zonas 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 zonas não pode afetar os conjuntos de dados que não foram adicionados à zona, 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 zonas instaladas:

- Um sistema de arquivos ZFS adicionado a uma zona não global deve ter a propriedade mountpoint definida como legado.
- Não adicione um conjunto de dados ZFS em uma zona não global quando a zona não global estiver configurada. Em vez disso, adicione um conjunto de dados ZFS depois que a zona for instalada.
- Quando o zonepath de origem e o zonepath de destino residirem em um sistema de arquivos do ZFS e estiverem no mesmo pool, o comando `zoneadm clone` utilizará automaticamente o ZFS para clonar uma zona. 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 zona. Para obter mais informações, consulte a [Parte II, “Zonas,” no Guia de administração do sistema: gerenciamento de recursos do Oracle Solaris Containers e Oracle Solaris Zones](#).
- Caso você delegue um sistema de arquivos do ZFS para uma zona não global, é preciso remover aquele sistema de arquivos da zona 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 zona 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 zona global. Um sistema de arquivos ZFS adicionado a uma zona não global deve ter a propriedade mountpoint definida como legado. Por exemplo, se o sistema de arquivos `tank/zone/zion` for adicionado a uma área não-global, defina a propriedade mountpoint na área global como a seguir:

```
# zfs set mountpoint=legacy tank/zone/zion
```

O sistema de arquivos ZFS pode ser adicionado a uma zona não global com o subcomando `add fs` do comando `zonecfg`.

No exemplo seguinte, um sistema de arquivos do ZFS é adicionado a uma zona não global por um administrador global na zona 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=/opt/data
zonecfg:zion:fs> end
```

Esta sintaxe adiciona o sistema de arquivos do ZFS, `tank/zone/zion`, à zona `zion` já configurada, montada em `/opt/data`. 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 zonas 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 zonas não pode alterar propriedades no sistema de arquivos, como `atime`, `readonly`, `compression`, etc. O administrador de zonas 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, “Zonas,” no Guia de administração do sistema: gerenciamento de recursos do Oracle Solaris Containers e Oracle Solaris Zones](#).

Delegando conjuntos de dados a uma zona não global

Se a prioridade for delegar a administração do armazenamento a uma zona, o ZFS oferece suporte à adição de conjuntos de dados a uma zona não global através do subcomando `add dataset` do comando `zonecfg`.

No exemplo a seguir, um sistema de arquivos ZFS é delegado a uma zona não global por um administrador de zonas globais na zona 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 zona `zion` já configurada. O administrador de zonas pode definir as propriedades do sistema de arquivos e criar sistemas de arquivos descendentes. Além disso, o administrador de zonas pode criar instantâneos, clones e, por outro lado, controlar toda a hierarquia do sistema de arquivos.

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 zona `zion` já configurada. Na zona `zion`, este sistema de arquivos não é acessível como `tank/zone/zion`, mas como um *pool virtual* com o nome `tank`. O alias do sistema de arquivos delegado fornece uma visão do pool original para a zona como um pool virtual. A propriedade do alias especifica o nome do pool virtual. Se nenhum alias for especificado, um alias padrão correspondente ao último componente do nome sistema de arquivos será usado. Se um alias específico não for fornecido, o alias padrão do exemplo acima não será `zion`.

Nos conjuntos de dados excluídos, o administrador de zonas pode definir as propriedades do sistema de arquivos e criar sistemas de arquivos descendentes. Além disso, o administrador de zonas pode criar instantâneos, clones e, por outro lado, controlar toda a hierarquia do sistema

de arquivos. Se os volumes ZFS forem criados nos sistemas de arquivos delegados, poderão ter conflito com volumes ZFS adicionados como recursos de dispositivos. Para obter mais informações, consulte a próxima seção.

Se estiver utilizando o Oracle Solaris Live Upgrade para atualizar o BE do ZFS com zonas 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 zonas, consulte [“Gerenciando propriedades do ZFS dentro de uma zona” na página 282](#).

Adicionando volumes ZFS a uma zona não global

Os volumes ZFS não podem ser adicionados a uma zona não global com o subcomando `add dataset` do comando `zonecfg`. No entanto, os volumes podem ser adicionados a uma zona com o subcomando `add device` do comando `zonecfg`.

No exemplo a seguir, um volume do ZFS é adicionado a uma zona não global por um administrador de zonas globais na zona 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` à zona `zion`.

Observe que a adição de um volume bruto a uma zona apresenta riscos de segurança implícitos, mesmo que o volume não corresponda a um dispositivo físico. Em particular, o administrador de zonas poderia criar sistemas de arquivos inválidos que causariam graves danos ao sistema quando houvesse uma tentativa de montagem. Para obter mais informações sobre a adição de dispositivos a zonas e os riscos de segurança relacionados, consulte [“Entendendo a propriedade zoned” na página 283](#).

Para obter mais informações sobre a adição de dispositivos a zonas, consulte a [Parte II, “Zonas,” no Guia de administração do sistema: gerenciamento de recursos do Oracle Solaris Containers e Oracle Solaris Zones](#).

Usando pools de armazenamento do ZFS dentro de uma zona

Os pools de armazenamento do ZFS não podem ser criados ou modificados dentro de uma zona. O modelo de administração delegado centraliza o controle dos dispositivos de armazenamento físico dentro da zona global e o controle do armazenamento virtual em zonas não globais. Enquanto um conjunto de dados no nível do pool puder ser adicionado a uma zona, nenhum comando que modifique as características físicas do pool, como criar, adicionar ou remover dispositivos, poderá ser realizado de dentro de uma zona. Mesmo se os dispositivos físicos forem adicionados a uma zona 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 zona.

Gerenciando propriedades do ZFS dentro de uma zona

Depois que o conjunto de dados for delegado a uma zona, o administrador de zonas poderá controlar propriedades específicas do conjunto de dados. Depois que um conjunto de dados é delegado a uma zona, 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 zona, cada conjunto de dados teria as seguintes propriedades.

Conjunto de dados	Visível	Gravável	Propriedades imutáveis
tank	Sim	Não	-
tank/home	Não	-	-
tank/data	Sim	Não	-
tank/data/matrix	Não	-	-
tank/data/zion	Sim	Sim	sharenfs, zoned, quota, reservation

Conjunto de dados	Visível	Gravável	Propriedades imutáveis
tank/data/zion/home	Sim	Sim	sharenfs, zoned

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 zonas não pode alterar a propriedade sharenfs porque as zonas não globais não podem agir como servidores NFS. O administrador de zonas não pode alterar a propriedade zoned porque isso apresentaria um risco de segurança, conforme descrito na próxima seção.

Os usuários privilegiados na zona podem alterar qualquer outra propriedade que possa ser definida, exceto as propriedades quota e reservation. Este comportamento permite que o administrador de zona global controle o consumo do espaço em disco de todos os conjuntos de dados utilizados pela zona não global.

Além disso, as propriedades sharenfs e mountpoint não podem ser alteradas pelo administrador de zonas globais depois que um conjunto de dados tiver sido delegado a uma zona não global.

Entendendo a propriedade zoned

Quando um conjunto de dados é delegado a uma zona não global, ele deve ser identificado com uma marca especial para que determinadas propriedades não sejam interpretadas dentro do contexto da zona global. Depois que o conjunto de dados tiver sido delegado a uma zona não global e estiver sob o controle de um administrador de zonas, 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 zona global. Além disso, a propriedade mountpoint não pode ser interpretada no contexto da zona global. Caso contrário, o administrador de zonas poderia afetar o espaço de nome da zona global. Para abordar este último, o ZFS usa a propriedade zoned para indicar que um conjunto de dados foi delegado a uma zona não global em um dado ponto no tempo.

A propriedade zoned é um valor booleano ativado automaticamente quando uma zona, contendo um conjunto de dados ZFS, é inicializada pela primeira vez. Um administrador de zonas não precisa ativar esta propriedade manualmente. Se a propriedade zoned estiver definida, o conjunto de dados não pode ser montado ou compartilhado na zona global. No exemplo a seguir, o tank/zone/zion foi delegado a uma zona, 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 zona ou quando uma zona 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 zona, 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 zona não global.

Usando pools de raiz alternativa do ZFS

Quando um pool é criado, ele é vinculado ao sistema host. O sistema host mantém informações sobre o pool para poder detectar quando o pool 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 pool 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 pool de raiz alternativa 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 pool de raiz alternativa 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 pool. Este sistema de arquivos poderá, então, ser importado e exportado através de um pool de raiz alternativa 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	ALTROOT
pool	44.8G	78K	44.7G	0%	ONLINE	/a

```
# zfs list pool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
pool	73.5K	44.1G	21K	/a/pool

Solução de problemas e recuperação de pools 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 contém as seguintes seções:

- “Identificando Problemas no ZFS” na página 287
- “Resolvendo Problemas Gerais de Hardware” na página 288
- “Identificando Problemas com Pools de Armazenamento do ZFS” na página 290
- “Resolvendo Problemas do Dispositivo de Armazenamento do ZFS” na página 295
- “Resolvendo Problemas do Sistema de Arquivos do ZFS” na página 308
- “Reparando uma configuração do ZFS danificada” na página 318
- “Reparando um sistema não inicializável” na página 318

Para obter informações sobre a recuperação completa do pool de raiz, consulte “Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 166.

Identificando Problemas no ZFS

Por ser uma combinação de um sistema de arquivos e um gerenciador de volumes, o ZFS pode exibir diferentes falhas. Este capítulo realça como diagnosticar falhas gerais de hardware e como resolver problemas do sistema de arquivos e do dispositivo do pool. Você poderá encontrar estes tipos de problemas:

- **Problemas Gerais de Hardware** – Os problemas de hardware podem afetar o desempenho do pool e a disponibilidade de dados do pool. Elimine os problemas gerais de hardware, como componentes e memória com falhas, antes de determinar problemas em um nível mais alto, como pools e sistemas de arquivos.
- **Problemas do Pool de Armazenamento de ZFS**
 - “Identificando Problemas com Pools de Armazenamento do ZFS” na página 290
 - “Resolvendo Problemas do Dispositivo de Armazenamento do ZFS” na página 295

- “Resolvendo Problemas do Sistema de Arquivos do ZFS” na página 308
- “Reparando uma configuração do ZFS danificada” na página 318
- “Reparando um sistema não inicializável” na página 318

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.

Resolvendo Problemas Gerais de Hardware

Examine as seções a seguir para determinar se os problemas do pool ou indisponibilidade do sistema de arquivos estão relacionados a um problema de hardware, como falha na placa do sistema, memória, dispositivo, HBA ou configuração incorreta.

Por exemplo, um disco com falhas em um pool de ZFS ocupado pode prejudicar muito o desempenho geral do sistema.

Se você começar diagnosticando e identificando problemas de hardware, que podem ser mais fáceis de detectar e verificar em todos os seus hardwares, poderá então passar para o diagnóstico do pool e problemas do sistema de arquivos, como descrito no restante deste capítulo. Se suas configurações de hardware, pool e sistema de arquivos estiverem íntegras, considere diagnosticar problemas de aplicativo, que geralmente são mais complexos de descobrir e não são abordados neste guia.

Identificando Falhas de Hardware e Dispositivo

O Solaris Fault Manager rastreia problemas de software, hardware e dispositivos específicos, identificando as informações de telemetria do erro que indicam um sintoma específico em um log de erros e reporta o diagnóstico real de falha quando o sintoma do erro resulta em uma falha concreta.

O comando a seguir identifica as falhas relacionadas a software ou hardware.

```
# fmadm faulty
```

Use o comando acima com frequência para identificar serviços ou dispositivos falhos.

Use o comando acima com frequência para identificar erros relacionados a hardware ou dispositivo.

```
# fmdump -eV | more
```

As mensagens de erro deste arquivo de log que descrevem os problemas `vdev.open_failed`, `checksum` ou `io_failure` precisam de sua atenção ou podem evoluir para falhas reais exibidas no comando `fmadm` que indica falha.

Se a afirmação acima indica que um dispositivo está danificado, é hora de verificar se existe um dispositivo sobressalente disponível.

Você também pode rastrear erros adicionais do dispositivo usando o comando `iostat`. Use a sintaxe a seguir para identificar um resumo das estatísticas de erro.

```
# iostat -en
---- errors ---
s/w h/w trn tot device
 0   0   0   0 c0t5000C500335F95E3d0
 0   0   0   0 c0t5000C500335FC3E7d0
 0   0   0   0 c0t5000C500335BA8C3d0
 0  12   0  12 c2t0d0
 0   0   0   0 c0t5000C500335E106Bd0
 0   0   0   0 c0t50015179594B6F11d0
 0   0   0   0 c0t5000C500335DC60Fd0
 0   0   0   0 c0t5000C500335F907Fd0
 0   0   0   0 c0t5000C500335BD117d0
```

Na saída acima, os erros são reportados em um disco interno `c2t0d0`. Use a sintaxe a seguir para exibir erros de dispositivo com mais detalhes.

```
# iostat -En
c0t5000C500335F95E3d0 Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE Product: ST930003SSUN300G Revision: 0B70 Serial No: 110672QFSB
Size: 300.00GB <300000000000 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
c0t5000C500335FC3E7d0 Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE Product: ST930003SSUN300G Revision: 0B70 Serial No: 110672TE67
Size: 300.00GB <300000000000 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
c0t5000C500335BA8C3d0 Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE Product: ST930003SSUN300G Revision: 0B70 Serial No: 110672SDF4
Size: 300.00GB <300000000000 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
c2t0d0 Soft Errors: 0 Hard Errors: 12 Transport Errors: 0
Vendor: AMI Product: Virtual CDROM Revision: 1.00 Serial No:
Size: 0.00GB <0 bytes>
Media Error: 0 Device Not Ready: 12 No Device: 0 Recoverable: 0
Illegal Request: 2 Predictive Failure Analysis: 0
```

Relatório de mensagens de erros do ZFS do sistema

Além de manter um rastreamento persistente de erros dentro do pool, o ZFS também exibe 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 em pools e em dispositivos**: se ocorrer uma falha em um pool ou em um dispositivo, o daemon do gerenciador de falhas relatará esses erros através de mensagens `syslog` bem como pelo comando `fmddump`.

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 pool. São normalmente o resultado de um problema do driver acompanhado por seu próprio conjunto de mensagens de erro.

Identificando Problemas com Pools de Armazenamento do ZFS

As seguintes seções descrevem como identificar e resolver problemas nos sistemas de arquivos ou nos pools de armazenamento do ZFS:

- [“Determinando se há problemas em um pool de armazenamento do ZFS” na página 291](#)
- [“Revisando a saída de `zpool status`” na página 292](#)
- [“Relatório de mensagens de erros do ZFS do sistema” na página 289](#)

Você pode usar os seguintes recursos para identificar problemas na configuração do seu ZFS:

- O pool de armazenamento do ZFS detalhado pode ser exibido através do comando `zpool status`.
- As falhas nos pools e dispositivos são relatadas por meio de mensagens de diagnóstico do ZFS/FMA.
- Os comandos anteriores do ZFS que modificaram as informações de estado do pool podem ser exibidos através 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 pool, embora existam vários problemas. Por exemplo, erros de dados corrompidos geralmente implicam que um dos dispositivos falhou, mas substituir os dispositivos danificados não deve resolver todos os problemas de corrupção de dados.

Além disso, um mecanismo de diagnóstico do ZFS gera diagnósticos e relata falhas de pools e dispositivos. Soma de verificação, E/S, dispositivo e erros de pool associados a essas falhas também são reportados. 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':
2012-11-12.13:01:31 zpool create tank mirror c0t1d0 c0t2d0 c0t3d0
2012-11-12.13:28:10 zfs create tank/eric
2012-11-12.13:37:48 zfs set checksum=off tank/eric
```

Nessa saída, note que as somas de verificação estão desativadas no sistema de arquivos `tank/eric`. 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 do comando `zpool status -x`.
- Reparar falhas envolve as seguintes etapas:
 - Substitua o dispositivo indisponível ou ausente e coloque-o 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 pool 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 pools que apresentam problemas. Se não houver um pool defeituoso no sistema, o comando exibirá 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 86](#).

Revisando a saída de `zpool status`

A saída de `zpool status` completa se assemelha ao ilustrado abaixo:

```
# zpool status tank
pool: tank
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
       the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
       see: http://www.sun.com/msg/ZFS-8000-2Q
       scan: scrub repaired 0 in 0h3m with 0 errors on Mon Nov 12 15:17:02 2012
config:

      NAME      STATE    READ WRITE CKSUM
      tank      DEGRADED     0     0     0
        mirror-0 DEGRADED     0     0     0
          c1t1d0 ONLINE       0     0     0
          c1t2d0 UNAVAIL       0     0     0  cannot open

errors: No known data errors
```

Essa saída é descrita na seção a seguir.

Informações gerais sobre o status do pool

Essa seção na saída de `zpool status` contém os seguintes campos, alguns dos quais são exibidos somente para os pools que apresentam problemas:

<code>pool</code>	Identifica o nome do pool.
<code>state</code>	Indica a integridade atual do pool. Esta informação se refere somente a capacidade que o pool apresenta de oferecer o nível de replicação necessário.
<code>status</code>	Descreve o que há de errado com o pool. 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.
erros	Identifica erros de dados ou ausência de erros de dados conhecidos.

Informações de configuração do pool de armazenamento do ZFS

O campo `config` na saída `zpool status` descreve a configuração dos dispositivos no pool, bem como seu estado e quaisquer erros gerados a partir dos dispositivos. O estado pode ser um dos seguintes : `ONLINE`, `FAULTED`, `DEGRADED` ou `SUSPENDED`. 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 298](#).

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 `UNAVAIL`, 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 307](#).

Status de Scrub do Pool de Armazenamento do ZSF

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 309](#).

Erros de Corrupimento de Dados do ZFS

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 pool. 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 `fmd` no console do sistema e no arquivo `/var/adm/messages`. Estas mensagens também podem ser rastreadas através do comando `fmdump`.

Para obter mais informações sobre a interpretação dos erros de corrupção de dados, consulte [“Identificando o tipo de corrupção de dados” na página 314.](#)

Resolvendo Problemas do Dispositivo de Armazenamento do ZFS

Examine as seções a seguir para resolver problema de dispositivo ausente, removido ou com falha.

Resolvendo um Dispositivo Ausente ou Removido

Se um dispositivo não puder ser aberto, será exibido como UNAVAILABLE na saída `zpool status`. Esse status significa que o ZFS não pôde abrir o dispositivo quando o pool 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 qualquer caso, o dispositivo apenas precisa ser reanexado ao sistema para restaurar as operações normais. Se você precisar substituir um dispositivo UNAVAIL devido a uma falha, consulte [“Substituindo um dispositivo em um pool de armazenamento do ZFS” na página 301.](#)

Se um dispositivo estiver UNAVAIL em um pool de raiz ou pool de raiz espelhado, consulte estas referências:

- **Falha no pool de raiz espelhado** – [“Inicializando a partir de um disco alternativo em um pool raiz ZFS espelhado” na página 159](#)
- **Substituindo um disco em um pool de raiz**
 - [“Como substituir um disco no pool raiz ZFS” na página 166](#)
- **Recuperação completa de desastres do pool de raiz** – [“Recuperando o pool raiz ZFS ou instantâneos do pool raiz” na página 166](#)

Por exemplo, uma mensagem do `cmd` 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: daleks
SOURCE: zfs-diagnosis, REV: 1.0
EVENT-ID: a1fb66d0-cc51-cd14-a835-961c15696fcb
DESC: The number of I/O errors associated with a ZFS device exceeded
acceptable levels. Refer to http://sun.com/msg/ZFS-8000-FD for more information.
AUTO-RESPONSE: The device has been offlined and marked as faulted. An attempt
```

will be made to activate a hot spare if available.
IMPACT: Fault tolerance of the pool may be compromised.
REC-ACTION: Run 'zpool status -x' and replace the bad device.

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
       scan: scrub repaired 0 in 0h0m with 0 errors on Tue Sep 27 16:59:07 2011
config:
```

NAME	STATE	READ	WRITE	CKSUM	
tank	DEGRADED	0	0	0	
mirror-0	DEGRADED	0	0	0	
c2t2d0	ONLINE	0	0	0	
c2t1d0	UNAVAIL	0	0	0	cannot open

errors: No known data errors

É possível observar nessa saída que o dispositivo `c2t1d0` não está funcionando. Se você determinar que o dispositivo está danificado, substitua-o.

Se necessário, utilize o comando `zpool online` para colocar o dispositivo substituído on-line. Por exemplo:

```
# zpool online tank c2t1d0
```

Informe ao FMA que o dispositivo foi substituído se a saída `fmadm faulty` identificar erro de dispositivo. Por exemplo:

```
# fmadm faulty
-----
TIME          EVENT-ID                                     MSG-ID        SEVERITY
-----
Sep 27 16:58:50 e6bb52c3-5fe0-41a1-9ccc-c2f8a6b56100  ZFS-8000-D3   Major

Host          : neo
Platform      : SUNW,Sun-Fire-T200      Chassis_id   :
Product_sn    :

Fault class   : fault.fs.zfs.device
Affects      : zfs://pool=tank/vdev=c75a8336cda03110
               faulted and taken out of service
Problem in   : zfs://pool=tank/vdev=c75a8336cda03110
               faulted and taken out of service

Description  : A ZFS device failed.  Refer to http://sun.com/msg/ZFS-8000-D3 for
               more information.
```



```
Response      : No automated response will occur.
Impact        : Fault tolerance of the pool may be compromised.
Action        : Run 'zpool status -x' and replace the bad device.
```

```
# fmadm repaired zfs://pool=tank/vdev=c75a8336cda03110
```

Como uma última etapa, confirme se o pool com o dispositivo substituído está íntegro. Por exemplo:

```
# zpool status -x tank
pool 'tank' is healthy
```

Resolvendo um Dispositivo Removido

Se um dispositivo for completamente removido do sistema, o ZFS detecta que o dispositivo não pode ser aberto e o coloca no estado REMOVED. Dependendo do nível de replicação dos dados do pool, essa remoção poderá ou não fazer com que todo o pool se torne indisponível. Se, em um dispositivo RAID-Z ou espelhado, um disco for removido, o pool continua acessível. Um pool pode se tornar UNAVAIL, o que significa que nenhum dado está acessível até que o dispositivo seja desanexado, sob as seguintes condições:

- 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

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 298](#).

Um pool pode ser SUSPENDED se a conectividade do dispositivo estiver comprometida. Um pool SUSPENDED permanece no estado de espera até que o problema do dispositivo seja resolvido. Por exemplo:

```
# zpool status cybermen
pool: cybermen
state: SUSPENDED
```

```
status: One or more devices are unavailable in response to IO failures.  
       The pool is suspended.  
action: Make sure the affected devices are connected, then run 'zpool clear' or  
       'fmadm repaired'.  
       see: http://www.sun.com/msg/ZFS-8000-HC  
       scan: none requested  
config:
```

NAME	STATE	READ	WRITE	CKSUM
cybermen	UNAVAIL	0	16	0
c8t3d0	UNAVAIL	0	0	0
c8t1d0	UNAVAIL	0	0	0

Depois da restauração da conectividade do dispositivo, limpe os erros de pool ou dispositivo.

```
# zpool clear cybermen  
# fmadm repaired zfs://pool=name/vdev=guid
```

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 pool já estava UNAVAIL ou SUSPENDED anteriormente ou se o sistema tiver sido reinicializado como parte do procedimento `attach`, o ZFS reexaminará todos os dispositivos ao tentar abrir o pool. Se o pool 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 73](#).

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 tank
pool: tank
state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
        entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scan: scrub in progress since Tue Sep 27 17:12:40 2011
      63.9M scanned out of 528M at 10.7M/s, 0h0m to go
      0 repaired, 12.11% done
config:
```

NAME	STATE	READ	WRITE	CKSUM
tank	ONLINE	2	0	0
mirror-0	ONLINE	2	0	0
c2t2d0	ONLINE	2	0	0
c2t1d0	ONLINE	2	0	0

errors: Permanent errors have been detected in the following files:

```
/tank/words
```

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 uma mensagem `syslog`, então provavelmente o dano é transitório.

O objetivo é responder à seguinte pergunta:

É provável que ocorra outro erro neste dispositivo?

Os erros que acontecem somente uma vez são considerados *transitórios* 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 transitórios, eles ainda poderão ter provocado erros de dados incorrigíveis no pool. 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 313](#).

Apagando Erros Transitórios do Dispositivo

Se os erros de dispositivo são considerados transitórios, 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 pool e quaisquer contagens de erros de dados associadas ao pool, utilize a seguinte sintaxe:

```
# zpool clear tank
```

Para obter mais informações sobre como eliminar erros do pool, consulte [“Limpendo erros de dispositivo do pool de armazenamento” na página 74](#).

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 301](#)
- [“Dispositivos que não podem ser substituídos” na página 302](#)
- [“Substituindo um dispositivo em um pool de armazenamento do ZFS” na página 302](#)
- [“Exibindo o status do resilvering” na página 307](#)

Determinando se um dispositivo pode ser substituído

Se o dispositivo a ser substituído faz parte de uma configuração redundante, devem existir réplicas suficientes para recuperar os dados válidos. Por exemplo, se dois discos em um espelho quadridimensional estiverem UNAVAIL, ambos os discos poderão ser substituídos porque há cópias em boas condições disponíveis. No entanto, se dois discos de um dispositivo virtual (raidz1) RAID-Z quadridirecional estiverem UNAVAIL, 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 UNAVAIL. No entanto, os dados corrompidos do dispositivo são copiados para o novo dispositivo, a menos que existam réplicas suficientes com dados bons.

Na seguinte configuração, o disco `c1t1d0` pode ser substituído e os dados do pool 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 UNAVAIL pode ser substituído. Os discos ONLINE também não podem ser substituídos porque o próprio pool está UNAVAIL

```
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 UNAVAIL, nenhuma substituição poderá ser efetuada porque o próprio pool poderá estar UNAVAIL.

Dispositivos que não podem ser substituídos

Se a perda de um dispositivo tornar o pool UNAVAIL ou se o dispositivo contiver muitos erros de dados em uma configuração não redundante, 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 pool e recriar a configuração e, em seguida, restaurar os dados a partir 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 316](#).

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 semelhante a esta:

```
# zpool replace tank c1t1d0 c2t0d0
```

Este comando migra os dados para o novo dispositivo a partir do dispositivo danificado ou de outros dispositivos do pool 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 75](#).

EXEMPLO 10-1 Substituindo um disco SATA em um pool de armazenamento do ZFS

O exemplo a seguir mostra como substituir um dispositivo (`c1t3d0`) no pool de armazenamento espelhado `tank` em um sistema com dispositivos SATA. Para substituir o disco `c1t3d0` por um novo disco no mesmo local (`c1t3d0`), desconfigure o disco antes de substituí-lo. Se o disco que será substituído não for SATA, consulte [“Substituindo dispositivos em um pool de armazenamento” na página 75](#).

A etapa básica segue:

EXEMPLO 10-1 Substituindo um disco SATA em um pool de armazenamento do ZFS *(Continuação)*

- Coloque o disco off-line (`c1t3d0`) para ser substituído. Não é possível desconfigurar um disco SATA que esteja sendo usado.
- Utilize o comando `cfgadm` para identificar o disco SATA (`c1t3d0`) para ser desconfigurado e `desconfigure-o`. O pool 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 UNAVAIL.
- Reconfigure o disco SATA (`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 pool para on, qualquer dispositivo novo, encontrado no mesmo local físico que um dispositivo que antes pertencia ao pool, será automaticamente formatado e substituído sem o uso do comando `zpool replace`. Este recurso pode não ser suportado em todos os hardwares.

- Se um disco danificado for substituído automaticamente por um disco de reserva, pode ser necessário desanexar o disco de reserva depois que o disco danificado for substituído. Por exemplo, se `c2t4d0` ainda é um disco de reserva ativo depois que o disco danificado foi substituído, desanexe-o.

```
# zpool detach tank c2t4d0
```

- Se o FMA estiver reportando o dispositivo com falha, você deverá apagar a falha do dispositivo.

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

O exemplo a seguir percorre as etapas para substituir um disco em um pool de armazenamento do ZFS.

```
# zpool offline tank c1t3d0
# cfgadm | grep c1t3d0
sata1/3::disk/c1t3d0          disk          connected    configured    ok
# cfgadm -c unconfigure sata1/3
Unconfigure the device at: /devices/pci@0,0/pci1022,7458@2/pci11ab,11ab@1:3
This operation will suspend activity on the SATA device
Continue (yes/no)? yes
# cfgadm | grep sata1/3
sata1/3                      disk          connected    unconfigured  ok
<Physically replace the failed disk c1t3d0>
# cfgadm -c configure sata1/3
# cfgadm | grep sata1/3
```

EXEMPLO 10-1 Substituindo um disco SATA em um pool de armazenamento do ZFS (Continuação)

```

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

```

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

EXEMPLO 10-1 Substituindo um disco SATA em um pool de armazenamento do ZFS *(Continuação)*

```

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 10-2 Substituindo um dispositivo de registro com falhas

O ZFS identifica falhas de log 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.

O exemplo a seguir mostra como recuperar de um dispositivo de log com falha `c0t5d0` no pool de armazenamento `pool`). A etapa básica segue:

- Reveja a saída `zpool status -x` e a mensagem de diagnóstico FMA, aqui descrita:

<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&alias=EVENT:ZFS-8000-K4>

- Substitua fisicamente o dispositivo de log com falha.
- Coloque o dispositivo de log on-line.
- Limpe a condição de erro do pool.
- Limpe o erro do FMA.

```

# zpool status -x
pool: pool

```

EXEMPLO 10-2 Substituindo um dispositivo de registro com falhas *(Continuação)*

```

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
```

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

```

# 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
```

```
# fmadm faulty
```

```
# fmadm repair zfs://pool=name/vdev=guid
```

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
```

- Reinicialize a condição de erro do dispositivo de log danificado.

EXEMPLO 10-2 Substituindo um dispositivo de registro com falhas (Continuação)

```
# zpool clear pool
```

Se quiser recuperar desse erro sem substituir o dispositivo de log danificado, poderá limpar o erro com o comando `zpool clear`. Nesse cenário, o pool irá operar em um modo degradado e os logs serão gravados no pool 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.

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 pool. 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 pool 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 e nem é possível criar um pool utilizando-o. O propósito desse dispositivo é exclusivamente exibir o progresso do resilvering e identificar qual dispositivo está sendo substituído.

Note que qualquer pool atualmente submetido ao resilvering é colocado no estado `ONLINE` ou `DEGRADED` porque o pool 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 pool está novamente `ONLINE` e o disco com falha original (`c1t0d0`) foi removido da configuração.

Resolvendo Problemas do Sistema de Arquivos do ZFS

Resolvendo Problemas de Dados em um Pool de Armazenamento do ZFS.

Exemplos de problemas de dados incluem:

- 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 substituiu 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 você substituiu 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.

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 pool 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 pool de armazenamento do ZFS são relacionados a falhas de hardware ou falhas de energia. Muitos problemas podem ser evitados através da utilização de pools redundantes. Se o pool for danificado por falhas de hardware ou queda de energia, consulte [“Reparando o dano de todo o pool de armazenamento do ZFS” na página 316](#).

Se o pool 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 todas 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 pool.

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 pool. 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 pool 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 Feb  2 12:54:00 2010
config:
  NAME          STATE          READ WRITE CKSUM
  tank          ONLINE         0     0     0
    mirror-0    ONLINE         0     0     0
      clt0d0    ONLINE         0     0     0
      clt1d0    ONLINE         0     0     0
```

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

Somente uma operação de scrubbing ativa por pool 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 scrub 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. Operações de scrub rotineiras 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 86](#).

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 pool. Se uma operação de scrub estiver em andamento, uma operação de resilver suspenderá a escovação atual e reiniciará depois da conclusão do polimento.

Para obter mais informações sobre o processo de resilver, consulte [“Exibindo o status do resilvering” na página 307](#).

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. Se for encontrado um erro no outro lado do espelho exatamente no mesmo local, ocorre corrompimento dos dados.

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 um processo de depuração rotineiro do pool, como explicado na seção a seguir. 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.

Resolvendo Problemas de Espaço do ZFS

Verifique as seções a seguir se não tiver certeza sobre como o ZFS reporta informações sobre espaço no sistema de arquivos e no pool. Além disso, consulte [“Contabilidade de espaço em disco do ZFS” na página 32](#).

Relatório de Espaço do Sistema de Arquivos ZFS

Os comandos `zpool list` e `zfs list` são melhores que os comandos `df` and `du` anteriores para determinar o espaço disponível no pool e no sistema de arquivos. Com os comandos legados, você não pode discernir facilmente entre o espaço do pool e do sistema de arquivos. Esses comandos legados também não contam espaço consumido pelos sistemas de arquivos ou instantâneos descendentes.

Por exemplo, o pool raiz a seguir (`rpool`) tem 5.46 GB alocados e 68.5 GB livres.

```
# zpool list rpool
NAME    SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool   74G   5.46G   68.5G   7%   1.00x  ONLINE  -
```

Se você comparar a contabilização do espaço do pool com a do sistema de arquivos examinando a coluna `USED` dos seus sistemas de arquivos individuais, poderá ver que o espaço do pool reportado em `ALLOC` é contabilizado no total de `USED` dos sistemas de arquivos. Por exemplo:

```
# zfs list -r rpool
NAME                                USED   AVAIL   REFER  MOUNTPOINT
rpool                              5.41G  67.4G   74.5K   /rpool
rpool/ROOT                         3.37G  67.4G   31K     legacy
rpool/ROOT/solaris                 3.37G  67.4G   3.07G   /
rpool/ROOT/solaris/var             302M   67.4G   214M   /var
rpool/dump                         1.01G  67.5G   1000M   -
rpool/export                       97.5K   67.4G   32K     /rpool/export
rpool/export/home                  65.5K   67.4G   32K     /rpool/export/home
rpool/export/home/admin            33.5K   67.4G   33.5K   /rpool/export/home/admin
rpool/swap                         1.03G  67.5G   1.00G   -
```

Relatórios de Espaço do Pool de Armazenamento do ZFS

O valor `SIZE` reportado pelo comando `zpool list` costuma ser a quantidade de espaço em disco físico do pool, mas varia dependendo do nível de redundância do pool. Consulte o exemplo abaixo. O comando `zfs list` lista o espaço utilizável disponível para os sistemas de arquivos, que é o espaço em disco menos a sobrecarga de metadados da redundância do pool do ZFS, se houver.

- **Pool de armazenamento não redundante** – Quando um pool é criado com um disco de 136 GB, o comando `zpool list` reporta valores iniciais de `SIZE` e `FREE` como 136 GB. O espaço `AVAIL` reportado pelo comando `zfs list` é de 134 GB devido à pequena quantidade de sobrecarga de metadados do pool. Por exemplo:

```
# zpool create tank c0t6d0
# zpool list tank
NAME    SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
tank    136G  95.5K   136G   0%   1.00x  ONLINE  -
# zfs list tank
NAME    USED   AVAIL   REFER  MOUNTPOINT
tank    72K   134G   21K     /tank
```


- **Pool de armazenamento espelhado** – Quando um pool é criado com um disco de 136 GB, o comando `zpool list` reporta SIZE como 136 GB e o valor inicial de FREE de 136 GB. Esse relatório é chamado de valor de espaço *deflated*. O espaço AVAIL reportado pelo comando `zfs list` é de 134 GB devido à pequena quantidade de sobrecarga de metadados do pool. Por exemplo:

```
# zpool create tank mirror c0t6d0 c0t7d0
# zpool list tank
NAME    SIZE  ALLOC   FREE      CAP  DEDUP  HEALTH  ALTROOT
tank    136G  95.5K   136G       0%  1.00x  ONLINE  -
# zfs list tank
NAME    USED  AVAIL  REFER  MOUNTPOINT
tank    72K   134G    21K    /tank
```

- **Pool de armazenamento RAID-Z** – Quando um pool `raidz2` é criado com um disco de 136 GB, o comando `zpool list` reporta SIZE como 408 GB e valores FREE iniciais como 408 GB. Esse relatório é chamado de valor de espaço em disco *inflado*, o que inclui sobrecarga de redundância, como informações sobre paridade. O espaço AVAIL inicial reportado pelo comando `zfs list` é de 133 GB devido à sobrecarga de redundância do pool. A discrepância de espaço entre a saída `zpool list` e a `zfs list` de um pool de RAID-Z ocorre porque `zpool list` reporta o espaço de pool inflado.

```
# zpool create tank raidz2 c0t6d0 c0t7d0 c0t8d0
# zpool list tank
NAME    SIZE  ALLOC   FREE      CAP  DEDUP  HEALTH  ALTROOT
tank    408G  286K   408G       0%  1.00x  ONLINE  -
# zfs list tank
NAME    USED  AVAIL  REFER  MOUNTPOINT
tank    73.2K  133G   20.9K    /tank
```

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 314
- “Reparando arquivos ou diretórios corrompidos” na página 315
- “Reparando o dano de todo o pool de armazenamento do ZFS” na página 316

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 pool não é redundante, se a corrupção ocorreu enquanto um pool 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 estiverem corrompidos, todo o pool ou partes do conjunto de dados se tornarão indisponíveis.
- 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 obter informações sobre como verificar a integridade dos dados do pool, consulte [“Verificando a integridade do sistema de arquivos ZFS” na página 309](#).

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 em todo o pool, o pool será colocado no estado **FAULTED** porque ele não poderá fornecer o nível solicitado 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. Ainda assim, é possível realizar a recuperação desta corrupção sem restaurar o pool inteiro.

Se o dano for dentro de um bloco de dados do arquivo, o arquivo poderá ser removido com segurança, 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 será exibido. Por exemplo:

```
/monkey/a.txt
```

- Se o caminho completo do arquivo for encontrado, mas o conjunto de dados não estiver montado, o nome do conjunto de dados será 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 pelo fato de o objeto não possuir 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 será exibido. Por exemplo:

```
monkey/dnode:<0x0>
```

- Se um objeto no metaobject set (MOS) está corrompido, então uma tag 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 os Dados Corrompidos com Várias Referências de Bloco

Se um sistema de arquivos danificado tiver dados corrompidos com várias referências de bloco, como instantâneos, o comando `zpool status -v` exibirá **todos** os caminhos de dados corrompidos. O relatório de `zpool status` atual de dados corrompidos é limitado pela quantidade de corrompimento de metadados e pelo fato de algum bloco ter sido reutilizado depois da execução do comando `zpool status`. Blocos com cancelamento de duplicação complicam mais a geração de relatórios de dados corrompidos.

Se você tiver dados corrompidos e o comando `zpool status -v` identificar que os dados do instantâneo foram afetados, considere executar o comando a seguir para identificar outros caminhos corrompidos:

Reparando o dano de todo o pool de armazenamento do ZFS

Se o dano estiver nos metadados do pool e o dano impedir que o pool seja aberto ou importado, as seguintes opções estarão disponíveis para você:

- Você pode tentar recuperar o pool usando o comando `zpool clear -F` ou o comando `zpool import -F`. Esses comandos tentam retornar as últimas transações do pool para um estado operacional. É possível utilizar o comando `zpool status` para rever um pool 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
clt1d0	ONLINE	0	0	2	
clt3d0	ONLINE	0	0	4	

O processo de recuperação, conforme descrito na saída anterior, deve usar o seguinte comando:

```
# zpool clear -F tpool
```

Se você tentar importar um pool de armazenamento danificado, irá visualizar mensagens semelhantes a estas:

```

# 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, conforme descrito na saída anterior, deve usar o seguinte comando:

```

# 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 pool danificado estiver no arquivo `zpool cache`, o problema será descoberto quando o sistema for inicializado e o pool danificado for reportado no comando `zpool status`. Se o pool não estiver no arquivo `zpool cache`, ele não será importado nem aberto com sucesso, e você verá mensagens de pools danificados quando tentar importar o pool.

- Você pode importar um pool danificado no modo somente leitura. Este método permite importar o pool de forma que você possa acessar os dados. Por exemplo:

```
# zpool import -o readonly=on tpool
```

Para obter mais informações sobre a importação de um pool somente leitura, consulte [“Importação de um pool no modo somente leitura” na página 104](#).

- Você pode importar um pool com um dispositivo de log ausente usando o comando `zpool import -m`. Para obter mais informações, consulte [“Importação de um pool com um dispositivo de log ausente” na página 102](#).

- Se o pool não pode ser recuperado pelo método de recuperação do pool, você deverá restaurar o pool e todos os seus dados de uma cópia de backup. O mecanismo utilizado varia muito dependendo da configuração do pool e da estratégia de backup. Primeiro, salve a configuração como exibida pelo comando `zpool status`, assim é possível recriá-la depois que o pool for destruído. Então, utilize o comando `zpool destroy -f` para destruir o pool. 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 uma configuração do ZFS danificada

O ZFS mantém um cache de pools 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 pool 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 pool (se estiver visível) e importando-o novamente.

Para obter mais informações sobre a importação e exportação de pools, consulte [“Migrando pools de armazenamento do ZFS” na página 97](#).

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 pool 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 pool 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 pools existentes no sistema, prevenindo-o de tentar acessar o pool defeituoso causando o problema. Você pode, então, passar para o estado normal do sistema usando o comando `svcadm milestone 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 pool 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 pool possui problemas utilizando o comando `fmddump -eV` para exibir os pools com erros fatais relatados.
- Importe os pools um por um, ignorando os pools que estão tendo problemas, como descrito na saída `fmddump`.

Práticas Recomendadas do Oracle Solaris ZFS

Este capítulo descreve práticas recomendadas para a criação, o monitoramento e a manutenção dos pools de armazenamento e sistemas de arquivos do ZFS.

Este capítulo traz as seguintes seções:

- “Práticas Recomendadas do Pool de Armazenamento” na página 321
- “Práticas Recomendadas do Sistema de Arquivos” na página 328

Práticas Recomendadas do Pool de Armazenamento

As seções a seguir oferecem práticas recomendadas para a criação e o monitoramento de pools de armazenamento do ZFS. Para obter mais informações sobre solução de problemas do pool de armazenamento, consulte o [Capítulo 10, “Solução de problemas e recuperação de pools do Oracle Solaris ZFS”](#).

Práticas Gerais do Sistema

- Mantenha o sistema atualizado com as versões e patches mais recentes do Solaris
- Confirme se o seu controlador cumpre os comandos de descarregamento de cache para conferir se seus dados foram gravados com segurança, o que é importante antes de alterar os dispositivos do pool ou dividir um pool espelhado. Em geral, isso não é problema para um hardware Oracle/Sun, mas é uma boa prática confirmar se a configuração de descarregamento de cache do hardware está ativada.
- Requisitos de tamanho de memória para uma carga de trabalho real do sistema
 - Com um volume de memória conhecido do aplicativo, como para aplicativo do banco de dados, você pode ocultar o tamanho do ARC para que o aplicativo não precise reclamar sua memória necessária no cache do ZFS.
 - Considere os requisitos de eliminação de duplicação de memória

- Identifique o uso de memória do ZFS com o seguinte comando:

```
# mdb -k
> ::memstat
Page Summary          Pages          MB  %Tot
-----
Kernel                388117          1516   19%
ZFS File Data          81321           317    4%
Anon                   29928           116    1%
Exec and libs          1359             5    0%
Page cache             4890             19    0%
Free (cachelist)       6030             23    0%
Free (freelist)        1581183         6176   76%

Total                  2092828         8175
Physical                2092827         8175
> $q
```

- Considere usar a memória de ECC para se proteger contra corrompimento de memória. O corrompimento silencioso da memória pode danificar seus dados.
- Execute backups regulares - Embora um pool criado com redundância de ZFS possa ajudar a reduzir o tempo de inatividade devido a falhas de hardware, ele não está imune a falhas de hardware, falhas de alimentação ou cabos desconectados. Faça backup de dados regularmente. Se seus dados forem importantes, faça backup deles. Estas são diferentes formas de fazer cópias de seus dados:
 - Instantâneos diários ou regulares do ZFS
 - Backups semanais dos dados do pool do ZFS. Você pode usar o comando `zpool split` para criar uma duplicata exata do pool de armazenamento espelhado do ZFS.
 - Backups mensais com o uso de um produto de backup de nível corporativo
- RAID de hardware
 - Considere usar o modo JBOD para matrizes de armazenamento em vez de RAID de hardware, de forma que o ZFS possa gerenciar o armazenamento e a redundância.
 - Use redundância de RAID ou ZFS, ou ambas
 - O uso de redundância do ZFS tem muitas vantagens - Para ambientes de produção, configure o ZFS para que ele possa corrigir inconsistência de dados. Use a redundância de ZFS, como RAID-Z, RAID-Z-2, RAID-Z-3, espelho, independentemente do nível de RAID implementado no dispositivo de armazenamento subjacente. Com essa redundância, as falhas do dispositivo de armazenamento subjacente ou suas conexões com o host podem ser descobertas e reparadas pelo ZFS.

Além disso, consulte [“Práticas de Criação de Pool nas Matrizes de Armazenamento Anexadas À Rede ou Locais”](#) na página 325.

- Os dumps de falha consomem mais espaço em disco, geralmente no tamanho 1/2 - 3/4 do intervalo de memória física.

Práticas de Criação do Pool de Armazenamento do ZFS

As seções a seguir oferecem práticas de pool gerais e mais específicas.

Práticas Gerais do Pool de Armazenamento

- Use discos inteiros para ativar o cache de gravação em disco e propiciar manutenção mais fácil. A criação de pools em segmentos adiciona complexidade ao gerenciamento e à recuperação do disco.
- Use a redundância de ZFS para que o ZFS possa reparar inconsistências de dados.
 - A mensagem a seguir é exibida quando um pool não redundante é criado:


```
# zpool create tank c4t1d0 c4t3d0
'tank' successfully created, but with no redundancy; failure
of one device will cause loss of the pool
```
 - Para pools espelhados, use pares de discos espelhados
 - Para pools de RAID-Z, agrupe os discos 3 - 9 por VDEV
 - Não combine componentes espelhados e RAID-S no mesmo pool. Esses pools são mais difíceis de gerenciar e pode haver problemas no desempenho.
- Use dispositivos de reserva para reduzir o tempo de inatividade por falhas de hardware
- Use discos de tamanho similares para ter equilíbrio e E/S nos dispositivos
 - LUNs menores podem ser expandidos para LUNs maiores
 - Não expanda os LUNs de tamanhos extremamente variados, como 128 MB a 2 TB, para manter os tamanhos ideais de metaslab
- Considere criar um pequeno pool de raiz e pools de dados maiores para suportar a recuperação mais rápida do sistema
- O tamanho mínimo recomendado para o pool é de 8 GB. Embora o tamanho mínimo do pool seja 64 MB, qualquer tamanho inferior a 8 GB pode dificultar a alocação e a solicitação de espaço livre no pool.
- O tamanho máximo de pool recomendado deve acomodar com conforto sua carga de trabalho ou tamanho dos dados. Não tente armazenar mais dados a ponto de não conseguir fazer os backups regulares. Caso contrário, seus dados ficarão em risco no caso de eventos imprevistos.

Práticas de Criação do Pool de Raiz

- Crie pools de raiz com segmentos usando o identificador s*. Não use o identificador p*. Em geral, um pool de raiz do ZFS do sistema é criado quando o sistema é instalado. Se você estiver criando um segundo pool de raiz ou recriando um pool de raiz, use a sintaxe semelhante à seguinte:

```
# zpool create rpool c0t1d0s0
```

Ou crie um pool de raiz espelhado. Por exemplo:

```
# zpool create rpool mirror c0t1d0s0 c0t2d0s0
```

- O pool raiz deve ser criado como uma configuração espelhada ou uma configuração de disco único. Nem configuração RAID-Z nem configuração com striping é suportada. 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`.
- Um pool raiz não pode possuir um dispositivo de log separado.
- É possível definir as propriedades do pool durante uma instalação de AI, mas o algoritmo de compactação `gzip` não é suportado nos pools de raiz.
- Não renomeie o pool raiz após sua criação por uma instalação inicial. Renomear o pool raiz pode resultar em um sistema não inicializável.
- Não crie um pool de raiz em um dispositivo USB para um sistema de produção, pois os discos do pool de raiz são críticos para a operação contínua, especificamente em um ambiente corporativo. Considere usar discos internos de um sistema para o pool de raiz, ou pelo menos use discos da mesma qualidade que os que você usaria para dados não raiz. Além disso, um dispositivo USB pode não ser grande o suficiente para suportar um tamanho de volume de dump equivalente a pelo menos metade do tamanho da memória física.

Práticas da Criação do Pool de Não Raiz

- Crie pools não raiz com discos inteiros usando o identificador `d*`. Não use o identificador `p*`.
 - O ZFS funciona melhor sem um software adicional de gerenciamento de volume.
 - Para obter melhor desempenho, use discos individuais ou pelo menos LUNs feitos de apenas alguns discos. Se o ZFS recebe mais visibilidade na configuração dos LUNs, pode tomar melhores decisões de agendamento de E/S.
 - Crie configurações de pool redundantes em vários controladores para reduzir o tempo de inatividade devido a uma falha no controlador.
 - **Pools de armazenamento espelhado** – Consome mais espaço em disco mas geralmente tem melhor desempenho com leituras aleatórias menores.

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

- **Pools de armazenamento de RAID-Z** – Pode ser criado com estratégias de 3 paridades, em que a paridade é igual a 1 (`raidz`), 2 (`raidz2`) ou 3 (`raidz3`). Uma configuração RAID-Z maximiza o espaço em disco e geralmente tem melhor desempenho quando os dados são gravados e lidos em blocos maiores (128 K ou mais).
 - Considere a configuração de um RAID-Z (`raidz`) único com 2 VDEVs de 3 discos (2+1) cada.

```
# zpool create rzpool raidz1 c1t0d0 c2t0d0 c3t0d0 raidz1 c1t1d0 c2t1d0 c3t1d0
```

- Uma configuração de RAIDZ-2 oferece melhor disponibilidade de dados e tem desempenho semelhante ao RAID-Z. O RAIDZ-2 tem MTDL (tempo médio de perda de dados) significativamente melhor que o RAID-Z ou espelhos bidirecionais. Crie uma configuração de RAID-Z (raidz2) de paridade dupla com 6 discos (4+2).

```
# zpool create rzpool raidz2 c0t1d0 c1t1d0 c4t1d0 c5t1d0 c6t1d0 c7t1d0
raidz2 c0t2d0 c1t2d0 c4t2d0 c5t2d0 c6t2d0 c7t2d0
```

- Uma configuração de RAIDZ-3 maximiza o espaço em disco e oferece excelente disponibilidade, pois pode suportar 3 falhas de disco. Crie uma configuração de RAID-Z (raidz2) de paridade dupla com 9 discos (6+3).

```
# zpool create rzpool raidz3 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0
c5t0d0 c6t0d0 c7t0d0 c8t0d0
```

Práticas de Criação de Pool nas Matrizes de Armazenamento Anexadas À Rede ou Locais

Considere as práticas de pool de armazenamento a seguir ao criar um pool de armazenamento do ZFS em uma matriz de armazenamento conectada local ou remotamente.

- Se você cria um pool em dispositivos SAN e a conexão de rede está lenta, os dispositivos do pool podem ficar UNAVAIL por um tempo. É necessário avaliar se a conexão de rede é apropriada para fornecer seus dados de maneira contínua. Além disso, considere que se você estiver usando dispositivos SAN para seu pool de raiz, talvez eles não fiquem disponíveis assim que o sistema for inicializado e os dispositivos do pool de raiz talvez fiquem UNAVAIL .
- Confirme com seu fornecedor de matrizes se a matriz de disco não está descarregando cache depois que o ZFS solicita o cache de gravação de descarga.
- Use discos inteiros, e não segmentos de discos, como dispositivos de pool de armazenamento, para que o Oracle Solaris ZFS ative os caches de discos pequenos locais que são descarregados nos momentos adequados.
- Para obter melhor resultado, crie um LUN para cada disco físico da matriz. O uso de apenas um LUN grande pode fazer com que o ZFS enfileire poucas operações de E/S para levar realmente o armazenamento ao desempenho ideal. Por outro lado, o uso de LUNs muito pequenos pode ter efeito de sobrecarregar o armazenamento com um número grande de operações pendentes de E/S.
- Uma matriz de armazenamento que usa software de provisionamento dinâmico (ou fino) para implementar alocação de espaço virtual não é recomendada para o Oracle Solaris ZFS. Quando o Oracle Solaris ZFS grava os dados modificados no espaço livre, ele grava todo o LUN. O processo de gravação do Oracle Solaris ZFS aloca todo o espaço virtual do ponto de vista da matriz de armazenamento, que nega o benefício de provisionamento dinâmico.

Considere que o software de provisionamento dinâmico pode ser desnecessário no uso do ZFS:

- Você pode expandir um LUN em um pool de armazenamento de ZFS existente e ele usará o novo espaço.

- Comportamento semelhante ocorre quando um LUN menor é substituído por um LUN maior.
- Se você avaliar as necessidades de armazenamento para seu pool e criar o pool com LUNs menores com as mesmas necessidades de armazenamento, poderá sempre expandir os LUNs para um tamanho maior, caso precise de mais espaço.
- Se a matriz puder apresentar dispositivos individuais (modo JBOD), considere criar pools redundantes de armazenamento de ZFS (espelho ou RAID-Z) nesse tipo de matriz, para que o ZFS possa reportar e corrigir inconsistências de dados.

Práticas de Criação de Pool para um Oracle Database

Considere as seguintes práticas de pool de armazenamento ao criar um banco de dados Oracle.

- Use um pool espelhado ou RAID de hardware para pools
- Os pools RAID-Z geralmente não são recomendados para cargas de trabalho aleatórias de leitura.
- Crie um pool pequeno separado com um dispositivo de log à parte para logs de redo do banco de dados.
- Crie um pool pequeno separado para o log de arquivamento

Para obter mais informações, consulte o documento a seguir:

http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle

Usando Pools de Armazenamento do ZFS no VirtualBox

- O Virtual Box foi configurado para ignorar os comandos de descarga de cache do armazenamento subjacente por padrão. Isso significa que no caso de uma falha no sistema ou de hardware, os dados podem ser perdidos.
- Ative a descarga de cache no Virtual Box emitindo este comando:

```
VBoxManage setextradata <VM_NAME> "VBoxInternal/Devices/<type>/0/LUN#<n>/Config/IgnoreFlush" 0
```

- <VM_NAME> é o nome da máquina virtual
- <type> é o tipo de controlador, piix3ide (se você estiver usando o controlador virtual IDE comum) ou ahci, se estiver usando um controlador SATA
- <n> é o número do disco

Práticas do Pool de Armazenamento para Desempenho

- Mantenha a capacidade abaixo de 80% para obter melhor desempenho

- Os pools espelhados são recomendados em pools de RAID-Z para cargas de trabalho aleatórias de leitura/gravação
- Dispositivos de log separados
 - Recomendado para aprimorar o desempenho síncrono de gravação
 - Uma carga de gravação síncrona alta impede a fragmentação de gravação de vários blocos no pool principal
- É recomendável separar os dispositivos de cache para melhorar o desempenho de leitura
- Scrub/resilver - Um pool RAID-Z muito grande, com vários dispositivos, terá operações mais longas de scrub e resilver
- O desempenho do pool está baixo – Use o comando `zpool status` para eliminar quaisquer problemas de hardware que prejudiquem o desempenho do pool. Se o comando `zpool status` não mostrar problemas, use o comando `fmddump` para exibir as falhas de hardware ou o comando `fmddump -ev` para revisar os erros de hardware que ainda não tenham gerado um relatório de falhas.

Práticas de Monitoramento e Manutenção do Pool de Armazenamento do ZFS

- Verifique se a capacidade do pool está abaixo de 80% para obter o melhor desempenho.
O desempenho do pool pode ser prejudicado quando ele está muito cheio e os sistemas de arquivos são atualizados frequentemente, como ocorre com um servidor de e-mails ocupado. Pools cheios podem causar problemas de desempenho, mas não outras falhas. Se a carga de trabalho principal for de arquivos imutáveis, mantenha o pool no intervalo de utilização de 95-96%. Mesmo com a maioria do conteúdo estático no intervalo de 95 a 96%, poderá haver problemas no desempenho de leitura, gravação e polimento.
 - Monitore o espaço do pool e do sistema de arquivos para verificar se eles não estão cheios.
 - Considere usar cotas e reservas para assegurar que o espaço do sistema de arquivos não ultrapasse a capacidade de pool de 80%.
- Monitore a integridade do pool
 - Em pools redundantes, monitore os pools com `zpool status` e `fmddump` semanalmente.
 - Em pools não redundantes, monitore os pools com `zpool status` e `fmddump` duas vezes por semana.
- Execute `zpool scrub` regularmente para identificar problemas de integridade de dados.
 - Se você tiver unidades de qualidade do consumidor, considere uma programa de scrub semanal.
 - Se você tiver unidades de qualidade do datacenter, considere uma programa de scrub mensal.

- Execute também uma operação de scrub antes de substituir dispositivos ou reduzir temporariamente uma redundância de pool para garantir que todos os dispositivos fiquem operacionais no momento.
- Monitorando falhas de pool ou dispositivos - Use `zpool status`, como descrito abaixo. Além disso, use `fmddump` ou `fmddump -eV` para verificar se ocorreram falhas ou erros nos dispositivos.
- Em pools redundantes, monitore a integridade dos pools com `zpool status` e `fmddump` semanalmente.
- Em pools não redundantes, monitore a integridade dos pools com `zpool status` e `fmddump` duas vezes por semana.
- O dispositivo do pool está **UNAVAIL** ou **OFFLINE** – Se um dispositivo de pool não estiver disponível, verifique se ele está listado na saída do comando `format`. Se o dispositivo não estiver listado na saída `format`, não ficará visível para o ZFS.

Se um dispositivo de pool estiver **UNAVAIL** ou **OFFLINE**, isso geralmente significa que o dispositivo falhou ou o cabo foi desconectado, ou algum outro problema de hardware, como cabo ou controlador danificado, tornou o dispositivo inacessível.

- Monitore o espaço do pool de armazenamento – Use o comando `zpool list` e o comando `zfs list` para identificar o quanto o disco foi consumido pelos dados do sistema de arquivos. Os instantâneos do ZFS podem consumir espaço em disco e se não estiverem listados pelo comando `zfs list`, podem consumir silenciosamente espaço em disco. Use o comando do instantâneo `zfs list -t` para identificar o espaço em disco consumido por instantâneos.

Práticas Recomendadas do Sistema de Arquivos

As seções a seguir descrevem as práticas recomendadas para o sistema de arquivos.

Práticas para Criação do Sistema de Arquivos

As seções a seguir descrevem as práticas de criação do sistema de arquivos do ZFS.

- Crie um sistema de arquivos por usuário para diretórios home.
- Considere usar cotas e reservas do sistema de arquivos para gerenciar e reservar espaço em disco para sistemas de arquivos importantes
- Considere usar cotas de usuário e grupo para gerenciar espaço em disco em um ambiente com vários usuários
- Use a herança de propriedade do ZFS para aplicar propriedades a vários sistemas de arquivos descendentes

Práticas de Criação de Sistema de Arquivos para um Oracle Database

Considere as seguintes práticas de sistemas de arquivos ao criar um banco de dados Oracle.

- Associe a propriedade `recordsize` do ZFS à `db_block_size` da Oracle.
- Crie uma tabela de banco de dados e indexe os sistemas de arquivos em um pool principal do banco de dados, usando um valor `recordsize` de 8 KB e o valor padrão `primarycache`.
- Crie dados temporários e desfaça sistemas de arquivos com `tablespace` no pool principal do banco de dados usando os valores padrão `recordsize` e `primarycache`.
- Crie o sistema de arquivos de log de arquivamento no pool de arquivamento, permitindo a compactação e o valor padrão `recordsize` e `primarycache` definido como `metadata`.

Para obter mais informações, consulte o documento a seguir:

http://blogs.oracle.com/storage/entry/new_white_paper_configuring_oracle

‘Práticas para Monitoramento do Sistema de Arquivos do ZFS

Monitore seus sistemas de arquivos do ZFS para garantir que eles fiquem disponíveis e para identificar problemas de consumo de espaço.

- Semanalmente, monitore a disponibilidade de espaço do sistema de arquivos com os comandos `zpool list` e `zfs list`, em vez de usar os comandos `du` e `df`, pois os comandos legados não atuam sobre o espaço consumido pelos sistemas de arquivos ou instantâneos descendentes.

Para obter mais informações, consulte “Resolvendo Problemas de Espaço do ZFS” na página 311.

- Exiba o consumo do espaço do sistema de arquivos usando o comando `zfs list -o space`.
- O espaço do sistema de arquivos pode ser consumido imperceptivelmente por instantâneos. É possível exibir todas as informações do conjunto de dados usando a seguinte sintaxe:

```
# zfs list -t all
```

- Um sistema de arquivos `/var` separado é criado automaticamente quando um sistema é instalado, mas você deve definir uma cota e reserva deste arquivo para garantir que ele não consuma espaço de pool de raiz sem que você perceba.
- Além disso, você pode usar o comando `fsstat` para exibir a atividade de operação do arquivo em sistemas de arquivos do ZFS. 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 /
new name   name attr attr lookup rddir  read read  write write
file remov chng  get  set   ops  ops  ops bytes ops bytes
832  589   286 837K 3.23K 2.62M 20.8K 1.15M 1.75G 62.5K 348M /
```

- Backups
 - Mantenha os instantâneos do sistema de arquivos
 - Considere o software de nível corporativo para backups semanais e mensais
 - Armazene instantâneos do pool de raiz em um sistema remoto para realizar a recuperação de metal

Descrição das versões do Oracle Solaris ZFS

Este apêndice descreve as versões disponíveis do ZFS, os recursos de cada versão e SO Solaris que oferece o recurso e a versão do ZFS.

As seguintes seções são fornecidas neste apêndice:

- [“Visão geral das versões do ZFS” na página 331](#)
- [“Versões do Pool do ZFS” na página 331](#)
- [“Versões do sistema de arquivos do ZFS” na página 333](#)

Visão geral das versões do ZFS

Novos recursos do sistema de arquivos e do pool 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 pool 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 pool e do sistema de arquivos.

Para obter informações sobre a utilização dos comandos `zpool upgrade` e `zfs upgrade`, consulte [“Atualização de sistemas de arquivos ZFS” na página 212](#) e [“Atualizando pools de armazenamento do ZFS” na página 106](#).

Versões do Pool do ZFS

A tabela a seguir fornece uma lista de versões do pool do ZFS disponíveis na versão do Oracle Solaris.

Versão	Solaris 10	Descrição
1	Solaris 10 6/06	Versão Inicial do ZFS
2	Solaris 10 11/06	Ditto blocks (metadados replicados)
3	Solaris 10 11/06	Discos de reserva com reposição em funcionamento e paridade dupla RAID-Z
4	Solaris 10 8/07	zpool histórico
5	Solaris 10 10/08	Algoritmo de compactação gzip
6	Solaris 10 10/08	Propriedade de pool 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 passsthrough-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	Retenções 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
23	Solaris 10 8/11	Slim ZIL
24	Solaris 10 8/11	Atributos do sistema
25	Solaris 10 8/11	Estatísticas de limpeza aprimoradas
26	Solaris 10 8/11	Melhor desempenho de exclusão de instantâneo
27	Solaris 10 8/11	Melhor desempenho de criação de instantâneo

Versão	Solaris 10	Descrição
28	Solaris 10 8/11	Várias substituições vdev
29	Solaris 10 8/11	Alocador híbrido RAID-Z/espelhamento
30	Solaris 10 1/13	Reservado
31	Solaris 10 1/13	Desempenho de <code>zfs list</code> aprimorado
32	Solaris 10 1/13	Tamanho de bloco de 1 MB

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 Oracle Solaris. Tenha em mente que um recurso disponível em uma versão de sistema de arquivos específica requer uma versão de pool específica.

Versão	Solaris 10	Descrição
1	Solaris 10 6/06	Versão do sistema de arquivos do ZFS inicial
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 <code>userquota</code> e <code>groupquota</code>
5	Solaris 10 8/11	Atributos do sistema

Índice

A

acessando

- instantâneo do ZFS
(exemplo de), 217

acinherit propriedade, 241

ACLs

- ACL no arquivo ZFS
descrição detalhada, 243

- ACL no diretório do ZFS
descrição detalhada, 244

- acinherit propriedade, 241

- definindo ACLs em arquivo ZFS (modo compacto)
(exemplo de), 256
descrição, 255

- definindo ACLs no arquivo ZFS (modo verboso)
descrição, 245

- definindo herança de ACL no arquivo ZFS (modo verboso)
(exemplo de), 249

- definindo nos arquivos ZFS
descrição, 242

- descrição, 235

- descrição do formato, 237

- diferenças das ACLs baseadas no esquema
POSIX, 236

- herança da ACL, 240

- modificando ACL trivial no arquivo ZFS (modo verboso)
(exemplo de), 246

- privilégios de acesso, 238

- propriedade ACL, 241

ACLs (*Continuação*)

- restaurando ACL comum no arquivo ZFS (modo verboso)
(exemplo de), 248

- sinalizadores de herança da ACL, 240

- tipos de entrada, 238

ACLs baseadas no esquema POSIX, descrição, 236

ACLs do NFSv4
descrição do formato, 237

ACLs do NFSv4, diferenças das ACLs baseadas no
esquema POSIX, 236

ACLs do NFSv4
herança da ACL, 240
modelo

- descrição, 235
sinalizadores de herança da ACL, 240

ACLs do Solaris
descrição do formato, 237
diferenças das ACLs baseadas no esquema
POSIX, 236
herança da ACL, 240
novo modelo

- descrição, 235
sinalizadores de herança da ACL, 240

ACLs NFSv4, propriedade ACL, 241

ACLs Solaris, propriedade ACL, 241

adicionando

- discos a uma configuração RAID-Z (exemplo de), 64

- dispositivo do log espelhado (exemplo de), 64

- dispositivos a um pool de armazenamento do ZFS
(zpool add)

adicionando, dispositivos a um pool de armazenamento do ZFS (`zpool add`) (*Continuação*)
 (exemplo de), 62
 dispositivos de cache (exemplo de), 66
 sistema de arquivos ZFS a uma zona não global (exemplo de), 279
 volume ZFS a uma zona não global (exemplo de), 281
administração delegada, visão geral, 261
administração delegada de ZFS, visão geral, 261
administração simplificada, descrição, 28
ajuste, tamanhos dos dispositivos de permuta e despejo, 156
alocado propriedade, descrição, 84
altroot propriedade, descrição, 84
anexando
 dispositivos a um pool de armazenamento do ZFS (`zpool attach`) (exemplo de), 67
apagando
 erros de dispositivo (`zpool clear`) (exemplo de), 300
armazenamento em pools, descrição, 25
arquivos, como componentes de pools de armazenamento do ZFS, 46
atime propriedade, descrição, 180
atualização
 pool de armazenamento do ZFS
 descrição, 106
atualização zfs, 212
atualizando
 sistemas de arquivos ZFS
 descrição, 212
autocorreção de dados, descrição, 49
autoreplace propriedade, descrição, 84

B

blocos de inicialização, instalando com `installboot` e `installgrub`, 159
bootfs propriedade, descrição, 84

C

cache file propriedade, descrição, 84
canmount propriedade
 descrição, 180
 descrição detalhada, 190
clone, definição, 28
clones
 criando (exemplo de), 221
 destruindo (exemplo de), 221
 recursos, 220
colocando dispositivos on-line e off-line
 pool de armazenamento do ZFS
 descrição, 72
colocando um dispositivo off-line (`zpool offline`)
 pool de armazenamento do ZFS
 (exemplo de), 73
colocando um dispositivo on-line
 pool de armazenamento do ZFS (`zpool online`) (exemplo de), 74
com zona propriedade, descrição, 187
compartilhando
 sistemas de arquivos ZFS
 descrição, 204
 exemplo de, 205
componentes de, pool de armazenamento do ZFS, 43
componentes do ZFS, requisitos para nomeação, 30
comportamento por espaço excedido, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 33
configuração espelhada
 descrição, 47
 recurso de redundância, 47
 visão conceitual, 47
configuração RAID-Z
 (exemplo de), 52
 paridade dupla, descrição, 47
 paridade única, descrição, 47
 recurso de redundância, 47
 visão conceitual, 47
configuração RAID-Z, adicionando discos a, (exemplo de), 64
configurando
 cota do sistema de arquivos ZFS (`zfs set quota`) exemplo de, 207

configurando (*Continuação*)
 pontos de montagem ZFS (`zfs set mountpoint`)
 (exemplo de), 201
 ZFS atime propriedade
 (exemplo de), 195
 conjunto, definição, 29
 conjunto de dados
 definição, 29
 descrição, 176
 conjuntos de permissões, definidos, 261
 contabilidade de espaço do ZFS, diferenças entre o ZFS
 e os sistemas de arquivos tradicionais, 32
 controlando, validação de dados (`scrubbing`), 310
 cotas e reservas, descrição, 206
 criação propriedade, descrição, 181
 criando
 clone do ZFS (exemplo de), 221
 hierarquia do sistema de arquivos ZFS, 39
 instantâneo do ZFS
 (exemplo de), 214
 pool de armazenamento com dispositivos de cache
 (exemplo de), 55
 pool de armazenamento de paridade dupla (`zpool
 create`)
 (exemplo de), 52
 pool de armazenamento de ZFS com dispositivos de
 log (exemplo de), 53
 pool de armazenamento do ZFS
 descrição, 50
 pool de armazenamento do ZFS (`zpool create`)
 (exemplo de), 50
 pool de armazenamento espelhado do ZFS (`zpool
 create`)
 (exemplo de), 50
 pool de armazenamento RAID-Z de paridade tripla
 (`zpool create`)
 (exemplo de), 52
 pool de armazenamento RAID-Z de paridade única
 (`zpool create`)
 (exemplo de), 52
 pools de raiz alternativa
 (exemplo de), 284
 sistema de arquivos ZFS, 40
 (exemplo de), 176

criando, sistema de arquivos ZFS (*Continuação*)
 descrição, 176
 um novo pool com a divisão de um pool de
 armazenamento espelhado (`zpool split`)
 (exemplo de), 69
 um pool de armazenamento do ZFS (`zpool eate`)
 (exemplo de), 37
 um sistema de arquivos do ZFS básico (`zpool
 create`)
 (exemplo de), 37
 volume ZFS
 (exemplo de), 275

D

dados
 corrompidos, 311
 corrupção identificada (`zpool status -v`)
 (exemplo de), 294
 reparação, 309
 resilvering
 descrição, 311
 scrubbing
 (exemplo de), 310
 validação (`scrubbing`), 310
 dados com soma de verificação, descrição, 27
 definindo
 ACLs em arquivo ZFS (modo compacto)
 (exemplo de), 256
 descrição, 255
 ACLs no arquivo ZFS (modo verboso)
 descrição, 245
 ACLs nos arquivos ZFS
 descrição, 242
 cota do ZFS
 (exemplo de), 195
 herança de ACL no arquivo ZFS (modo verboso)
 (exemplo de), 249
 propriedade `compression`
 (exemplo de), 40
 propriedade `mountpoint`, 40
 propriedade `quota` (exemplo de), 41
 propriedade `sharenfs`
 (exemplo de), 40

definindo (*Continuação*)

- reserva do sistema de arquivos ZFS
 - (exemplo de), 210

delegação propriedade, descrição, 85
delegando

- conjunto de dados a uma zona não global
 - (exemplo de), 280
- permissões (exemplo de), 266

- delegando permissões, `zfs allow`, 265

- delegando permissões a um grupo, (exemplo de), 267

- delegando permissões a um usuário individual,
 - (exemplo de), 266

desanexando

- dispositivos de um pool de armazenamento do ZFS
 - (`zpool detach`)
 - (exemplo de), 69

descompartilhando

- sistemas de arquivos ZFS
 - exemplo de, 205

desmontando

- sistemas de arquivos ZFS
 - (exemplo de), 204

- desmontar `zfs`, (exemplo de), 204

- despejo de memória, salvando, 158

destruindo

- clone do ZFS (exemplo de), 221
- instantâneo do ZFS
 - (exemplo de), 215
- pool de armazenamento do ZFS
 - descrição, 50
- pool de armazenamento do ZFS (`zpool destroy`)
 - (exemplo de), 61
- sistema de arquivos ZFS
 - (exemplo de), 177
- sistema de arquivos ZFS com dependentes
 - (exemplo de), 178

detectando

- dispositivos em uso
 - (exemplo de), 58
- níveis de replicação inconsistentes
 - (exemplo de), 59

determinando

- se um dispositivo pode ser substituído
 - descrição, 301

determinando (*Continuação*)

- tipo de falha do dispositivo
 - descrição, 298

diferenças entre o ZFS e os sistemas de arquivos tradicionais

- comportamento por espaço excedido, 33
- contabilidade de espaço do ZFS, 32
- gerenciamento de volume tradicional, 34
- granularidade do sistema de arquivos, 31
- montando sistemas de arquivos ZFS, 34
- novo modelo de ACL do Solaris, 34

discos, como componentes de pools de armazenamento do ZFS, 44

discos de reserva

- descrição de
 - (exemplo de), 78

discos inteiros, como componentes de pools de armazenamento do ZFS, 44

- disponível propriedade, descrição, 180

dispositivo de log espelhado, adicionando, (exemplo de), 64

dispositivo virtual, definição, 30

dispositivos de cache

- considerações para utilizar, 55
- criando um pool de armazenamento do ZFS com
 - (exemplo de), 55

- dispositivos de cache, adicionando, (exemplo de), 66

- dispositivos de cache, removendo, (exemplo de), 66

- dispositivos de intercâmbio e de dump, descrição, 154

- dispositivos de log espelhados, criando um pool de armazenamento de ZFS (exemplo de), 53

- dispositivos de log separados, considerações para uso, 53

- dispositivos de permuta e de despejo, problemas, 155

- dispositivos de permuta e despejo, ajuste dos tamanhos de, 156

dispositivos em uso

- detectando
 - (exemplo de), 58

dispositivos virtuais, como componentes de pools de armazenamento do ZFS, 57

distribuição dinâmica

- descrição, 49
- recurso do pool de armazenamento, 49

dividindo um pool de armazenamento espelhado
 (zpool split)
 (exemplo de), 69
 dumpadm, ativando um dispositivo de despejo, 158

E

enviando e recebendo
 dados do sistema de arquivos ZFS
 descrição, 223
 espelho, definição, 29
 estado zpool -x, (exemplo de), 94
 exibindo
 estado de integridade detalhado do pool de
 armazenamento do ZFS
 (exemplo de), 95
 estado de integridade do pool de armazenamento do
 ZFS
 (exemplo de), 94
 estado de integridade dos pools de armazenamento
 descrição de, 92
 estatísticas de E/S de pool de armazenamento do ZFS
 descrição, 90
 estatísticas de E/S de todos os pools de
 armazenamento do ZFS
 (exemplo de), 91
 estatísticas de E/S de vdev do pool de
 armazenamento do ZFS
 (exemplo de), 91
 permissões delegadas (exemplo de), 270
 relatório de registro do sistema das mensagens de
 erro do ZFS
 descrição, 289
 exportando
 pool de armazenamento do ZFS
 (exemplo de), 99

F

failmode propriedade, descrição, 85
 falhas, 287

G

gerenciamento de volume tradicional, diferenças entre
 o ZFS e os sistemas de arquivos tradicionais, 34
 granularidade do sistema de arquivos, diferenças entre
 o ZFS e os sistemas de arquivos tradicionais, 31

H

health, descrição, 85
 herdando
 propriedades ZFS (zfs inherit)
 descrição, 196
 hierarquia do sistema de arquivos, criando, 39

I

identificando
 pool de armazenamento ZFS para importar (zpool
 import -a)
 (exemplo de), 99
 requisitos de armazenamento, 37
 tipo de corrupção de dados zpool status -v)
 (exemplo de), 314
 importando
 pool de armazenamento do ZFS
 (exemplo de), 102
 pool de armazenamento do ZFS a partir de diretórios
 alternativos (zpool import -d)
 (exemplo de), 101
 pools de raiz alternativa
 (exemplo de), 285
 inicializando
 sistema de arquivos raiz, 159
 um BE do ZFS com boot -L e boot -Z em sistemas
 SPARC, 161
 instalação inicial do sistema de arquivos raiz ZFS,
 (exemplo de), 114
 Instalação JumpStart
 sistema de arquivos de raiz
 exemplos de perfil, 128
 instalação JumpStart
 sistema de arquivos raiz
 problemas, 129

instalando

- sistema de arquivo raiz ZFS
 - requisitos, 111

- sistema de arquivos raiz ZFS
 - recursos, 110

- sistema de arquivos ZFS
 - instalação JumpStart, 126

- instalando blocos de inicialização
 - installboot e installgroup (exemplo de), 159

installing

- ZFS root file system
 - (initial installation), 114

instantâneo

- acessando
 - (exemplo de), 217
- contagem de espaço, 218
- criando
 - (exemplo de), 214
- definição, 30
- destruindo
 - (exemplo de), 215
- recursos, 213
- renomeando
 - (exemplo de), 216
- revertendo
 - (exemplo de), 219

L

legenda EFI

- descrição, 44
- interação com o ZFS, 44

limpando

- um dispositivo em um pool de armazenamento do ZFS (zpool clear)
 - descrição, 74

limpando um dispositivo

- pool de armazenamento do ZFS)
 - (exemplo de), 75

lista

- informações do pool do ZFS, 38
- pools de armazenamento do ZFS
 - descrição, 86

lista (*Continuação*)

- sistemas de arquivos do ZFS (zfs list)
 - (exemplo de), 41

listando

- descendentes dos sistemas de arquivos do ZFS
 - (exemplo de), 193

- pools de armazenamento do ZFS
 - (exemplo de), 87

- propriedades do ZFS (zfs list)
 - (exemplo de), 196

- propriedades do ZFS por valor de origem
 - (exemplo de), 198

- propriedades ZFS para scripting
 - (exemplo de), 199

- sistemas de arquivos ZFS
 - (exemplo de), 192

- sistemas de arquivos ZFS sem informações de cabeçalho
 - (exemplo de), 194

- tipos de sistemas de arquivos ZFS
 - (exemplo de), 194

listshares propriedade, descrição, 86

listsnapshots propriedade, descrição, 86

livre propriedade, descrição, 85

luactivate

- sistema de arquivos raiz
 - (exemplo de), 133

lucreate

- BE do ZFS de um BE do ZFS
 - (exemplo de), 135

- migração de sistema de arquivos raiz
 - (exemplo de), 132

M

migrando

- sistemas de arquivos raiz UFS para sistemas de arquivos raiz ZFS
 - (Oracle Solaris Live Upgrade), 129
 - problemas, 131

- migrando pools de armazenamento do ZFS,
 - descrição, 98

- modelo de ACL, Solaris, diferenças entre o ZFS e os sistemas de arquivos tradicionais, 34

modificando
 ACL trivial no arquivo ZFS (modo verboso)
 (exemplo de), 246
 modo de propriedade da ACL, `aclinherit`, 180
 modo de propriedade de ACL, `aclmode`, 180
 modos de falha
 dados corrompidos, 311
 dispositivo ausente (UNAVAIL), 297
 dispositivos danificados, 308
 montado propriedade, descrição, 182
 montando
 sistemas de arquivos ZFS
 (exemplo de), 202
 montando sistemas de arquivos ZFS, diferenças entre o
 ZFS e os sistemas de arquivos tradicionais, 34

N

níveis de replicação inconsistentes
 detectando
 (exemplo de), 59
 notificando
 ZFS sobre dispositivo reanexado (`zpool online`)
 (exemplo de), 298

O

Oracle Solaris Live Upgrade
 migração de sistema de arquivos raiz
 (exemplo de), 132
 para migração de sistemas de arquivos raiz, 129
 problemas de migração de sistemas de arquivos de
 raiz, 131

P

pacote de fluxos
 recursivos, 225
 replicação, 224
 pacote de fluxos de replicação, 224
 pacote de fluxos recursivos, 225

palavras-chave de perfil do JumpStart, sistema de
 arquivos raiz ZFS, 126
 perfis de direitos, para gerenciamento de sistemas de
 arquivos e pools de armazenamento do ZFS, 35
 ponto de montagem
 padrão para pools de armazenamento ZFS, 60
 padrão para sistema de arquivos ZFS, 176
 pontos de montagem
 automático, 200
 gerenciando ZFS
 descrição, 200
 legado, 200
 pool de armazenamento do ZFS
 adicionando dispositivos a (`zpool add`)
 (exemplo de), 62
 versões
 descrição, 331
 pool de armazenamento espelhado (`zpool create`),
 (exemplo de), 50
 pools de armazenamento de ZFS, falhas, 287
 pools de armazenamento do ZFS
 anexando dispositivos a (`zpool attach`)
 (exemplo de), 67
 apagando erros de dispositivo (`zpool clear`)
 (exemplo de), 300
 atualizando
 descrição, 106
 colocando dispositivos on-line e off-line
 descrição, 72
 colocando um dispositivo off-line (`zpool offline`)
 (exemplo de), 73
 componentes, 43
 configuração espelhada, descrição, 47
 configuração RAID-Z, descrição, 47
 conjunto
 definição, 29
 corrupção de dados identificada (`zpool status -v`)
 (exemplo de), 294
 criando (`zpool create`)
 (exemplo de), 50
 criando configuração espelhada (`zpool create`)
 (exemplo de), 50
 criando uma configuração RAID-Z (`zpool create`)
 (exemplo de), 52

pools de armazenamento do ZFS (*Continuação*)

- dados corrompidos
 - descrição, 311
- desanexando dispositivos de (`zpool detach`)
 - (exemplo de), 69
- destruindo (`zpool destroy`)
 - (exemplo de), 61
- determinando o tipo de falha do dispositivo
 - descrição, 298
- determinando se há problemas (`zpool status -x`)
 - descrição, 292
- determinando se um dispositivo pode ser substituído
 - descrição, 301
- dispositivo ausente (UNAVAIL)
 - descrição, 297
- dispositivo virtual
 - definição, 30
- dispositivos danificados
 - descrição, 308
- dispositivos virtuais, 57
- distribuição dinâmica, 49
- dividindo um pool de armazenamento espelhado (`zpool split`)
 - (exemplo de), 69
- espelho
 - definição, 29
- estatísticas de E/S de vdev
 - (exemplo de), 91
- estatísticas de todos os pools
 - (exemplo de), 91
- exibindo estado de integridade, 92
 - (exemplo de), 94
- exibindo estado de integridade detalhado
 - (exemplo de), 95
- exportando
 - (exemplo de), 99
- identificando o tipo de corrupção de dados (`zpool status -v`)
 - (exemplo de), 314
- identificando problemas
 - descrição, 291
- importando
 - (exemplo de), 102

pools de armazenamento do ZFS (*Continuação*)

- importando de diretórios alternativos (`zpool import -d`)
 - (exemplo de), 101
- informações gerais sobre o status do pool para solução de problemas
 - descrição, 292
- limpando um dispositivo
 - (exemplo de), 75
- listando
 - (exemplo de), 87
- mensagens de erro do sistema
 - descrição, 289
- migrando
 - descrição, 98
- notificando ZFS sobre dispositivo reanexado (`zpool online`)
 - (exemplo de), 298
- perfis de direitos, 35
- pools de raiz alternativa, 284
- processo de visualização de resilvering
 - (exemplo de), 307
- RAID-Z
 - definição, 29
- realizando uma simulação (`zpool create -n`)
 - (exemplo de), 60
- recuperando um pool destruído
 - (exemplo de), 105
- reparação de dados
 - descrição, 309
- reparando arquivos ou diretórios corrompidos
 - descrição, 315
- reparando o dano de todo o pool
 - descrição, 318
- reparando um sistema não inicializável
 - descrição, 318
- reparando uma configuração ZFS danificada, 318
- resilvering
 - definição, 30
- resilvering e scrubbing de dados
 - descrição, 311
- script de saída de pool de armazenamento
 - (exemplo de), 88

- pools de armazenamento do ZFS (*Continuação*)
 - scrubbing de dados
 - (exemplo de), 310
 - descrição, 310
 - substituindo um dispositivo (zpool replace)
 - (exemplo de), 75, 302
 - substituindo um dispositivo ausente
 - (exemplo de), 295
 - usando arquivos, 46
 - usando discos inteiros, 44
 - validação de dados
 - descrição, 310
- pools de armazenamento do ZFS (zpool online)
 - colocando um dispositivo on-line
 - (exemplo de), 74
- pools de armazenamento ZFS
 - identificando para importar (zpool import -a)
 - (exemplo de), 99
 - ponto de montagem padrão, 60
- pools de raiz alternativa
 - criando
 - (exemplo de), 284
 - descrição, 284
 - importando
 - (exemplo de), 285
- primarycache propriedade, descrição, 182
- propriedade capacity, descrição, 85
- propriedade compression, descrição, 181
- propriedade compressratio, descrição, 181
- propriedade copies, descrição, 181
- propriedade de pool do ZFS, cachefile, 84
- propriedade delegation, desativando, 262
- propriedade devices, descrição, 181
- propriedade do pool do ZFS, health, 85
- propriedade exec, descrição, 182
- propriedade guid, descrição, 85
- propriedade mountpoint, descrição, 182
- propriedade origin, descrição, 183
- propriedade quota, descrição, 183
- propriedade recordsize, descrição, 183
- propriedade referenced, descrição, 183
- propriedade sharenfs
 - descrição, 185, 204
- propriedade size, descrição, 86
- propriedade snapdir, descrição, 185
- propriedade type, descrição, 185
- propriedade volsize, descrição, 186
- propriedade zoned, descrição detalhada, 283
- propriedades configuráveis de ZFS
 - com zona, 187
 - refquota, 184
 - reserva, 184
 - soma de verificação, 181
- propriedades configuráveis do ZFS
 - aclmode, 180
 - atime, 180
 - canmount
 - descrição detalhada, 190
 - descrição, 188
 - primarycache, 182
 - recordsize
 - descrição detalhada, 190
 - refreservation, 184
 - secondarycache, 184
 - setuid, 185
 - shareiscsi, 185
 - somente leitura, 183
 - utilizadas
 - descrição detalhada, 188
 - versão, 186
 - volblocksize, 186
 - volsize
 - descrição detalhada, 191
 - xattr, 187
- propriedades de conjunto ZFS, listsnapshots, 86
- propriedades de pool do ZFS, failmode, 85
- propriedades de somente leitura do ZFS
 - criação, 181
 - montado, 182
 - usedbydataset, 186
 - usedbyrefreservation, 186
 - utilizado, 185
- propriedades de ZFS
 - aclmode, 180
- Propriedades de ZFS
 - com zona, 187
- propriedades de ZFS
 - descrição, 179

propriedades de ZFS (*Continuação*)

descrição de propriedades herdáveis, 179

Propriedades de ZFS

refquota, 184

reserva, 184

propriedades de ZFS

secondarycache, 182

Propriedades de ZFS

secondarycache, 184

soma de verificação, 181

propriedades definíveis de ZFS

aclinherit, 180

devices, 181

mountpoint, 182

recordsize, 183

sharenfs, 185

snapdir, 185

volsize, 186

propriedades definíveis do ZFS

canmount, 180

compression, 181

copies, 181

exec, 182

quota, 183

propriedades do grupo ZFS, delegação, 85

propriedades do pool de ZFS, listshare, 86

propriedades do pool do ZFS

alocado, 84

alroot, 84

autoreplace, 84

bootfs, 84

capacity, 85

guid, 85

livre, 85

size, 86

versão, 86

propriedades do usuário do ZFS

(exemplo de), 191

descrição detalhada, 191

propriedades do ZFS

aclinherit, 180

canmount, 180

compression, 181

compressratio, 181

propriedades do ZFS (*Continuação*)

configuráveis, 188

copies, 181

descrição, 179

devices, 181

exec, 182

gerenciamento dentro de uma zona

descrição, 282

herdável, descrição de, 179

mountpoint, 182

origin, 183

propriedade zoned

descrição detalhada, 283

quota, 183

recordsize, 183

referenced, 183

sharenfs, 185

snapdir, 185

somente leitura, 187

type, 185

Propriedades do ZFS, usedbychildren, 185

propriedades do ZFS

volsize, 186

propriedades somente leitura de ZFS

compressratio, 181

origin, 183

referenced, 183

type, 185

usedbychildren, 185

propriedades somente leitura do ZFS

descrição, 187

disponível, 180

propriedades ZFS

atime, 180

available, 180

canmount

descrição detalhada, 190

criação, 181

montado, 182

propriedades do usuário

descrição detalhada, 191

Propriedades ZFS

recordsize

descrição detalhada, 190

propriedades ZFS
 refreservation, 184
 setuid, 185
 shareiscsi, 185
 somente leitura, 183
 utilizadas
 descrição detalhada, 188
 utilizado, 185
 versão, 186
 volblocksize, 186

Propriedades ZFS
 volsize
 descrição detalhada, 191
propriedades ZFS
 xattr, 187

R

RAID-Z, definição, 29
read-only properties of ZFS, usedbysnapshots, 186
recebendo
 dados do sistema de arquivos ZFS (zfs receive)
 (exemplo de), 227
recordsize propriedade, descrição detalhada, 190
recuperando
 pool de armazenamento destruído do ZFS
 (exemplo de), 105
recursos de replicação de ZFS, espelhado ou
 RAID-Z, 47
refquota propriedade, descrição, 184
refreservation propriedade, descrição, 184
removendo, dispositivos de cache (exemplo de), 66
removendo permissões, zfs unallow, 266
renomeando
 instantâneo do ZFS
 (exemplo de), 216
 sistema de arquivos ZFS
 (exemplo de), 178
reparando
 dano de todo o pool
 descrição, 318
 reparando arquivos ou diretórios corrompidos
 descrição, 315

reparando (*Continuação*)
 um sistema não inicializável
 descrição, 318
 uma configuração ZFS danificada
 descrição, 318
requisitos, para a instalação e o Live Upgrade, 111
requisitos de armazenamento, identificando, 37
requisitos de hardware e software, 36
requisitos para nomeação, componentes do ZFS, 30
reserva
 criando
 (exemplo de), 78
reserva propriedade, descrição, 184
resilvering, definição, 30
resilvering e scrubbing de dados, descrição, 311
restaurando
 ACL
 comum no arquivo ZFS (modo verboso), 248
revertendo
 instantâneo do ZFS
 (exemplo de), 219

S

salvando
 dados do sistema de arquivos do ZFS (zfs send)
 (exemplo de), 226
 desejos de memória
 savecore, 158
savecore, salvando desejos de memória, 158
script
 saída de pool de armazenamento do ZFS
 (exemplo de), 88
scrubbing
 (exemplo de), 310
 validação de dados, 310
secondarycache propriedade, descrição, 184
semânticas transacionais, descrição, 26
setting
 legacy mount points
 (exemplo de), 202
setuid propriedade, descrição, 185
shareiscsi propriedades, descrição, 185

simulação

- criação de pool de armazenamento do ZFS (zpool create -n)

- (exemplo de), 60

sistema de arquivos, definição, 29

sistema de arquivos ZFS

- descrição, 175

- listando propriedades para scripting

- (exemplo de), 199

- versões

- descrição, 331

sistemas de arquivo do ZFS

- modificando ACL trivial no arquivo ZFS (modo verboso)

- (exemplo de), 246

sistemas de arquivo ZFS

- criando um volume ZFS volume

- (exemplo de), 275

- requisitos de instalação e Live Upgrade, 111

sistemas de arquivos do ZFS

- administração simplificada

- descrição, 28

- armazenamento em pools

- descrição, 25

- clone, 221

- clones

- definição, 28

- conjunto de dados

- definição, 29

- delegando conjunto de dados a uma zona não global

- (exemplo de), 280

- descrição, 25

- perfis de direitos, 35

- requisitos para nomeação de componentes, 30

- salvando fluxos de dados (zfs send)

- (exemplo de), 226

- semânticas transacionais

- descrição, 26

- sistema de arquivos

- definição, 29

- utilizando em um sistema Solaris com zonas

- instaladas

- descrição, 278

sistemas de arquivos do ZFS (*Continuação*)

- volume

- definição, 30

sistemas de arquivos raiz ZFS

- migração de sistemas de arquivos raiz com o Oracle

- Solaris Live Upgrade

- (exemplo de), 132

sistemas de arquivos ZFS

- ACL no arquivo ZFS

- descrição detalhada, 243

- ACL no diretório do ZFS

- descrição detalhada, 244

- adicionando sistema de arquivos ZFS a uma zona

- não global

- (exemplo de), 279

- adicionando volume ZFS a uma zona não global

- (exemplo de), 281

- atualizando

- descrição, 212

- clone

- substituindo um sistema de arquivos por

- (exemplo de), 222

- clones

- descrição, 220

- compartilhando

- descrição, 204

- exemplo de, 205

- configurando atime propriedade

- (exemplo de), 195

- contagem de espaço do instantâneo, 218

- criando

- (exemplo de), 176

- dados com soma de verificação

- descrição, 27

- definindo a propriedade quota

- (exemplo de), 195

- definindo ACLs em arquivo ZFS (modo compacto)

- (exemplo de), 256

- descrição, 255

- definindo ACLs no arquivo ZFS (modo verboso)

- descrição, 245

- definindo ACLs nos arquivos ZFS

- descrição, 242

sistemas de arquivos ZFS (*Continuação*)

- definindo herança de ACL no arquivo ZFS (modo verboso)
 - (exemplo de), 249
- definindo uma reserva
 - (exemplo de), 210
- definir ponto de montagem (`zfs set mountpoint`)
 - (exemplo de), 201
- descompartilhando
 - exemplo de, 205
- desmontando
 - (exemplo de), 204
- destruindo
 - (exemplo de), 177
- destruindo com dependentes
 - (exemplo de), 178
- dispositivos de intercâmbio e de dump
 - descrição, 154
- dispositivos de permuta e de despejo
 - problemas, 155
- dispositivos de permuta e despejo
 - ajuste dos tamanhos de, 156
- enviando e recebendo
 - descrição, 223
- gerenciamento dentro de uma zona
 - descrição, 282
- gerenciando pontos de montagem
 - descrição, 200
- gerenciando pontos de montagem automáticos, 200
- gerenciando pontos de montagem de legado
 - descrição, 200
- herdando propriedade de (`zfs inherit`)
 - (exemplo de), 196
- inicializando um BE do ZFS com `boot -L` e `boot -Z`
 - (exemplo de SPARC), 161
- inicializando um sistema de arquivos raiz
 - descrição, 159
- instalação JumpStart do sistema de arquivos raiz, 126
- instalando um sistema de arquivos raiz, 110
- instantâneo
 - acessando, 217
 - criando, 214
 - definição, 30

sistemas de arquivos ZFS, instantâneo (*Continuação*)

- descrição, 213
- renomeando, 216
- revertendo, 219
- instantâneos
 - destruindo, 215
- listando
 - (exemplo de), 192
- listando descendentes
 - (exemplo de), 193
- listando propriedades de (`zfs list`)
 - (exemplo de), 196
- listando propriedades por valor de origem
 - (exemplo de), 198
- listando sem informações de cabeçalho
 - (exemplo de), 194
- listando tipos de
 - (exemplo de), 194
- migração de sistemas de arquivos raiz com o Oracle Solaris Live Upgrade, 129
- montando
 - (exemplo de), 202
- ponto de montagem padrão
 - (exemplo de), 176
- problemas de migração de sistemas de arquivos raiz, 131
- recebendo fluxos de dados (`zfs receive`)
 - (exemplo de), 227
- renomeando
 - (exemplo de), 178
- restaurando ACL comum no arquivo ZFS (modo verboso)
 - (exemplo de), 248
- setting legacy mount point
 - (exemplo de), 202
- soma de verificação
 - definição, 28
- tipos de conjunto de dados
 - descrição, 193
- sistemas de arquivos ZFS (`zfs set quota`)
 - configurando uma cota
 - exemplo de, 207

solução de problemas

- apagar erros de dispositivo(`zpool clear`)
 - (exemplo de), 300
- corrupção de dados identificada (`zpool status -v`)
 - (exemplo de), 294
- determinando o tipo de corrupção de dados (`zpool status -v`)
 - (exemplo de), 314
- determinando o tipo de falha do dispositivo
 - descrição, 298
- determinando se há problemas (`zpool status -x`), 292
- determinando se um dispositivo pode ser substituído
 - descrição, 301
- dispositivo ausente (UNAVAIL), 297
- dispositivos danificados, 308
- falhas de ZFS, 287
- identificando problemas, 291
- informações gerais sobre o status do pool
 - descrição, 292
- notificando ZFS sobre dispositivo reanexado (`zpool online`)
 - (exemplo de), 298
- relatório de registro do sistema das mensagens de erro, 289
- reparando arquivos ou diretórios corrompidos
 - descrição, 315
- reparando o dano de todo o pool
 - descrição, 318
- reparando um sistema não inicializável
 - descrição, 318
- reparando uma configuração ZFS danificada, 318
- substituindo um dispositivo (`zpool replace`)
 - (exemplo de), 302, 307
- substituindo um dispositivo ausente
 - (exemplo of), 295
- soma de verificação, definição, 28
- soma de verificação propriedade, descrição, 181
- somente leitura propriedade, descrição, 183
- substituindo
 - um dispositivo (`zpool replace`)
 - (exemplo de), 75, 302, 307
 - um dispositivo ausente
 - (exemplo de), 295

T

terminologia

- clone, 28
- conjunto de dados, 29
- dispositivo virtual, 30
- espelho, 29
- instantâneo, 30
- pool, 29
- RAID-Z, 29
- resilvering, 30
- sistema de arquivos, 29
- soma de verificação, 28
- volume, 30
- tipos de conjunto de dados, descrição, 193

U

- usedbychildren propriedade, descrição, 185
- usedbydataset propriedade, descrição, 186
- usedbyreservation propriedade, descrição, 186
- usedbysnapshots propriedade, descrição, 186
- utilizadas propriedades, descrição detalhada, 188
- utilizado propriedade, descrição, 185

V

- verificar, integridade dos dados ZFS, 309
- versão do ZFS
 - recurso do ZFS e SO Solaris
 - descrição, 331
- versão propriedade, descrição, 186
- versão propriedade, descrição, 86
- volblocksize propriedade, descrição, 186
- volsize propriedade, descrição detalhada, 191
- volume, definição, 30
- volume ZFS, descrição, 275

X

- xattr propriedade, descrição, 187

Z

zfs allow
 descrição, 265
 exibindo permissões delegadas, 270
zfs create
 (exemplo de), 40, 176
 descrição, 176
zfs destroy, (exemplo de), 177
zfs destroy -r, (exemplo de), 178
ZFS file systems, initial installation of ZFS root file system, 114
zfs get, (exemplo de), 196
zfs get -H -o, (exemplo de), 199
zfs get -s, (exemplo de), 198
zfs inherit, (exemplo de), 196
ZFS intent log (ZIL), descrição, 53
zfs list
 (exemplo de), 41, 192
zfs list -H, (exemplo de), 194
zfs list -r, (exemplo de), 193
zfs list -t, (exemplo de), 194
zfs mount, (exemplo de), 202
zfs promote, promoção de clone (exemplo de), 222
ZFS properties
 usedbydataset, 186
 usedbyrefreservation, 186
 usedbysnapshots, 186
zfs receive, (exemplo de), 227
zfs rename, (exemplo de), 178
zfs send, (exemplo de), 226
zfs set atime, (exemplo de), 195
zfs set compression, (exemplo de), 40
zfs set mountpoint
 (exemplo de), 40, 201
zfs set mountpoint=legacy, (exemplo de), 202
zfs set quota
 (exemplo de), 41
zfs set quota, (exemplo de), 195
zfs set quota
 exemplo de, 207
zfs set reservation, (exemplo de), 210
zfs set sharenfs, (exemplo de), 40
zfs set sharenfs=on, exemplo de, 205
zfs unallow, descrição, 266

zonas

adicionando o sistema de arquivos ZFS a uma zona não global
 (exemplo de), 279
 adicionando volume ZFS a uma zona não global
 (exemplo de), 281
 delegando conjunto de dados a uma zona não global
 (exemplo de), 280
 gerenciamento de propriedades do ZFS dentro de uma zona
 descrição, 282
 propriedade zoned
 descrição detalhada, 283
 utilizando com sistemas de arquivos do ZFS
 descrição, 278
zpool add, (exemplo de), 62
zpool attach, (exemplo de), 67
zpool clear
 (exemplo de), 75
 descrição, 74
zpool create
 (exemplo de), 37, 38
 conjunto básico
 (exemplo de), 50
 pool de armazenamento espelhado
 (exemplo de), 50
 pool de armazenamento RAID-Z
 (exemplo de), 52
zpool create -n, simulação (exemplo de), 60
zpool destroy, (exemplo de), 61
zpool detach, (exemplo de), 69
zpool export, (exemplo de), 99
zpool import -a, (exemplo de), 99
zpool import -D, (exemplo de), 105
zpool import -d, (exemplo de), 101
zpool import nome, (exemplo de), 102
zpool iostat, todos os pools (exemplo de), 91
zpool iostat -v, vdev (exemplo de), 91
zpool list
 (exemplo de), 38, 87
 descrição, 86
zpool list -Ho name, (exemplo de), 88
zpool offline, (exemplo de), 73
zpool online, (exemplo de), 74

zpool replace, (exemplo de), 75
zpool split, (exemplo de), 69
zpool status -v, (exemplo de), 95
zpool upgrade, 106