

Oracle® Solaris ゾーンの紹介

ORACLE®

Part No: E54011-02
2014 年 12 月

Copyright © 2004, 2014, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ, AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

| | |
|---|----------|
| このドキュメントの使用方法 | 7 |
| 1 Oracle Solaris ゾーンの概要 | 9 |
| ゾーンの概要 | 10 |
| このリリースでサポートされるゾーン | 11 |
| 不変ゾーン | 11 |
| ipkg ゾーンの solaris ゾーンへの変換について | 12 |
| このリリースでのゾーンブランド | 12 |
| Oracle Solaris カーネルゾーン | 12 |
| デフォルトの Oracle Solaris ゾーン | 13 |
| Oracle Solaris 10 ゾーン | 13 |
| ゾーンブランドの概要 | 14 |
| Oracle Solaris Trusted Extensions システム上での Oracle Solaris ゾーンの使用 | 14 |
| Oracle Solaris クラスタのゾーンクラスタ | 14 |
| ブランドゾーンについて | 15 |
| ブランドゾーンで実行中のプロセス | 16 |
| ゾーンを使用する場合 | 16 |
| ゾーンのしくみ | 18 |
| ゾーンの機能別のサマリー | 19 |
| 非大域ゾーンの管理のしくみ | 20 |
| 非大域ゾーンの作成のしくみ | 21 |
| 非大域ゾーンの状態モデル | 21 |
| 非大域ゾーンの特性 | 25 |
| 非大域ゾーンでのリソース管理機能の使用 | 25 |
| ゾーン関連の SMF サービス | 25 |
| 非大域ゾーンのモニタリング | 26 |
| 非大域ゾーンによって提供される機能 | 26 |
| このリリースの Oracle Solaris ゾーンについて | 28 |
| ライブゾーン再構成 | 31 |

| | |
|--|----|
| 2 非大域ゾーンの構成の概要 | 33 |
| ゾーンのリソースについて | 33 |
| ゾーン管理での権利プロファイルと役割の使用 | 34 |
| zonecfg template プロパティ | 34 |
| インストール前の構成処理 | 35 |
| ゾーンのコンポーネント | 35 |
| ゾーンの名前とパス | 36 |
| ゾーンの自動ブート | 36 |
| 不変ゾーンの file-mac-profile プロパティ | 36 |
| admin リソース | 36 |
| dedicated-cpu リソース | 37 |
| solaris-kz のみ: virtual-cpu リソース | 38 |
| capped-cpu リソース | 38 |
| スケジューリングクラス | 39 |
| 物理メモリの制御と capped-memory リソース | 40 |
| solaris および solaris10 のみ: rootzpool リソース | 41 |
| zpool リソースの自動追加 | 43 |
| ゾーンネットワークインタフェース | 44 |
| ゾーンでマウントされるファイルシステム | 50 |
| ファイルシステムのマウントと更新 | 51 |
| ゾーン内のホスト ID | 52 |
| 非大域ゾーンの /dev ファイルシステム | 52 |
| 非大域ゾーンのリムーバブル lofi デバイス | 52 |
| 非大域ゾーン内でのディスク形式のサポート | 53 |
| ストレージ URI を持つカーネルゾーンのデバイスリソース | 53 |
| 構成可能な特権 | 54 |
| リソースプールの関連付け | 54 |
| ゾーン規模のリソース制御の設定 | 55 |
| ゾーンのコメントの追加 | 58 |
| zonecfg コマンドの使用 | 58 |
| zonecfg のモード | 59 |
| zonecfg の対話型モード | 59 |
| zonecfg のコマンドファイルモード | 62 |
| ゾーン構成データ | 62 |
| リソースタイプとプロパティ | 62 |
| リソースタイプのプロパティ | 68 |
| Tecla コマンド行編集ライブラリ | 78 |

| | |
|-----------|----|
| 用語集 | 79 |
| 索引 | 83 |

このドキュメントの使用方法

- 概要 – ゾーンのテクノロジーと、ゾーンで使用可能なリソースについて説明します。
- 対象読者 – システム管理者、技術者、および認定サービスプロバイダ。
- 必要な知識 – ネットワーク構成とリソース割り当ての知識を含む、Oracle Solaris オペレーティングシステムの経験。

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (<http://www.oracle.com/pls/topic/lookup?ctx=E56342>) に含まれています。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

◆◆◆ 第 1 章

Oracle Solaris ゾーンの概要

Oracle Solaris オペレーティングシステムの Oracle™ Solaris ゾーン機能では、システム上にアプリケーションを実行するための隔離された環境が提供されます。

この章では、ゾーンの概要について説明します。

また、この章では、次の一般的なゾーンについての各トピックについても説明しています。

- 10 ページの「ゾーンの概要」
- 11 ページの「このリリースでサポートされるゾーン」
- 28 ページの「このリリースの Oracle Solaris ゾーンについて」
- 11 ページの「不変ゾーン」
- 12 ページの「ipkg ゾーンの solaris ゾーンへの変換について」
- 16 ページの「ゾーンを使用する場合」
- 18 ページの「ゾーンのしくみ」
- 26 ページの「非大域ゾーンによって提供される機能」
- 12 ページの「このリリースでのゾーンブランド」

次の章では、ゾーン構成のリソースとプロパティについて説明します。

システムでゾーンの作成を開始する準備ができている場合、『[Oracle Solaris カーネルゾーンの作成と使用](#)』および『[Oracle Solaris ゾーンの実行と使用](#)』にスキップします。Oracle Solaris 10 が実行されており、そこに非大域 native ゾーンを含めることができるシステムを Oracle Solaris 11 システム上のゾーンに移行する準備ができている場合、『[Oracle Solaris 10 ゾーンの実行と使用](#)』を参照してください。

注記 - Oracle Solaris Trusted Extensions システム上でゾーンを使用する方法については、『[Trusted Extensions 構成と管理](#)』の第 13 章「Trusted Extensions でのゾーンの管理」を参照してください。

ゾーンの概要

Oracle Solaris ゾーンパーティション技術は、オペレーティングシステムサービスを仮想化して、アプリケーションを実行するための隔離されたセキュアな環境を提供するために使用します。非大域ゾーンは、ゾーンと呼ばれ、Oracle Solaris オペレーティングシステムの 1 つのインスタンス内で作成される仮想化されたオペレーティングシステム環境です。オペレーティングシステムのインスタンスは大域ゾーンと呼ばれます。Oracle Solaris カーネルゾーンは、SRU (Support Repository Update)、またはホストのものとは異なるカーネルバージョンを実行できます。

仮想化の目標は、個々のデータセンターコンポーネントの管理からリソースのプールの管理に移行することです。サーバーを正しく仮想化すると、サーバー使用効率が向上し、サーバー資産をより効率的に使用できるようになります。個々のシステムの隔離を維持するサーバー統合プロジェクトで成功を収めるためにも、サーバーの仮想化は重要です。

仮想化は、複数のホストおよびサービスを単一のマシン上に統合する必要がある場合に推進されます。仮想化すると、ハードウェア、インフラストラクチャー、および管理を共有することにより、コストを削減できます。次のような利点があります。

- ハードウェアの使用効率の向上
- より柔軟なリソース割り当て
- 電力要件の削減
- 管理コストの低下
- 保有コストの低減
- システムのアプリケーション間の管理およびリソースの境界

ゾーンを作成すると、そのアプリケーション実行環境で実行されるプロセスは、システムのほかの部分から隔離されます。この分離を行うことで、1 つのゾーン内で稼働しているプロセスがほかのゾーンで稼働しているプロセスをモニタリングしたりそれらのプロセスに影響を及ぼしたりすることが防止されます。root 資格で実行されているプロセスであっても、ほかのゾーンの活動を表示したり影響を与えたりすることはできません。ハードウェアリソースを同時に共有しながら、1 サーバーあたりに 1 つのアプリケーションという配備モデルを維持するには、Oracle Solaris ゾーンを使用します。

また、ゾーンにより、アプリケーションを配備するマシンの物理的属性からアプリケーションを分離する抽象層も提供されます。このような属性の例として、物理デバイスパスがあります。

ゾーンは、Oracle Solaris 10 または Oracle Solaris 11 リリースを稼働しているマシン上で使用できます。システム上での solaris ゾーンおよび solaris10 ゾーンの数の上限は 8192

です。1 つのシステムで効率的にホストできるゾーン数は、すべてのゾーンで実行されるアプリケーションソフトウェアに必要な総リソース量、およびそのシステムのサイズによって決まります。これらの概念については、『Oracle Solaris ゾーンの作成と使用』の第 1 章「非大域ゾーンの計画および構成方法」で説明されています。

Oracle Solaris カーネルゾーンを実行している場合のこれらの概念の詳細は、『Oracle Solaris カーネルゾーンの作成と使用』の「Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件」を参照してください。

このリリースでサポートされるゾーン

単一ホストの大域ゾーン内で実行されている非大域 solaris および solaris10 ブランドゾーンは、Oracle Solaris 11.2 リリースのサポート対象プラットフォームとして定義されているすべてのアーキテクチャーでサポートされます。

Oracle Solaris カーネルゾーンは、T4+ と M5+ SPARC マシン、Nehalem+ Intel マシン、および Barcelona+ AMD マシンで実行できます。カーネルゾーンのシステム要件については、『Oracle Solaris カーネルゾーンの作成と使用』の「Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件」を参照してください。

不変ゾーン

不変ゾーンは、読み取り専用ルートを持つ solaris ゾーンです。大域ゾーンと非大域ゾーンの両方を不変ゾーンにできます。読み取り専用ゾーンは、file-mac-profile プロパティを設定することによって構成できます。複数の構成が利用できます。読み取り専用ゾーンルートは、安全な実行時の境界を拡張します。

Oracle Solaris 不変大域ゾーンは、不変ゾーン機能を大域ゾーンに拡張しました。不変ゾーンと不変カーネルゾーンでは、トラステッドパスログインは、zlogin コマンド `zlogin(1)` を使用して呼び出すことができます。

`zonecfg add dataset` を使用して追加のデータセットが指定されたゾーンは、それらのデータセットに対する完全な制御権を引き続き持ちます。ファイルシステムが読み取り専用を設定されていないかぎり、`zonecfg add fs` を使用して追加のファイルシステムが指定されたゾーンは、それらのファイルシステムに対する完全な制御権を引き続き持ちます。

詳細は、『Oracle Solaris ゾーンの実装と使用』の第 12 章「不変ゾーンの実装と管理」を参照してください。

ipkg ゾーンの実装と solaris ゾーンへの変換について

Oracle Solaris 11 Express リリースのお客様をサポートするために、ipkg ゾーンとして構成されたゾーンは solaris ゾーンに変換され、Oracle Solaris 11.2 に対する pkg update または zoneadm attach の実行時に solaris として報告されます。ipkg の名前がゾーンの実装時に使用された場合、その名前が solaris の名前にマップされます。Oracle Solaris 11 Express ホストからエクスポートされた zonecfg ファイルのインポートがサポートされます。

zonecfg info や zoneadm list -v などのコマンドの出力には、Oracle Solaris 11.2 システム上のデフォルトのゾーンに対して solaris というブランドが表示されます。

このリリースでのゾーンブランド

Oracle Solaris カーネルゾーン

Oracle Solaris カーネルゾーン機能は、ゾーン内の完全なカーネルおよびユーザー環境を提供し、さらにホストとゾーンの間でカーネルの分離を強化します。ブランド名は solaris-kz です。カーネルゾーンは、既存のツールである zonecfg、zoneadm、および zlogin を使用して大域ゾーンから管理されます。カーネルゾーンの管理者は、デフォルトの solaris ゾーン管理者より柔軟にゾーンを構成および管理できます。たとえば、大域ゾーンにインストールされているパッケージに制限されることなく、カーネルバージョンを含め、ゾーンのインストール済みパッケージを完全に更新および変更できます。ゾーン専用ストレージの管理、ZFS プールの作成と破棄、および iSCSI と CIFS の構成を行うことができます。カーネルゾーン内に solaris ゾーンをインストールして、階層 (入れ子にされた) ゾーンを生成できます。カーネルゾーンでは一時停止および再開がサポートされます。ソースマシンでゾーンを一時停止して、ターゲットマシンでゾーンを再開することで、カーネルゾーンを移行できます。

Oracle Solaris カーネルゾーンを使用するには、パッケージ brand-solaris-kz をシステムにインストールする必要があります。使用しているマシンがカーネルゾーンをサポートするかどうかを判別するには、28 ページの「このリリースの Oracle Solaris ゾーンについて」を参照

してください。Oracle Solaris 11.2 がインストールされている場合、マシンで `virtinfo` コマンドを実行することもできます。Oracle Solaris カーネルゾーンの詳細は、『[Oracle Solaris カーネルゾーンの作成と使用](#)』および [solaris-kz\(5\)](#) のマニュアルページを参照してください。`virtinfo` コマンドの詳細は、『[Oracle Solaris カーネルゾーンの作成と使用](#)』の「[ホストでカーネルゾーンのサポートを確認する方法](#)」および [virtinfo\(1M\)](#) のマニュアルページを参照してください。

デフォルトの Oracle Solaris ゾーン

Oracle Solaris ゾーン機能は、アプリケーションの完全な実行時環境です。ゾーンは、アプリケーションからプラットフォームリソースへの仮想マッピングを提供します。ゾーンを使用すると、Oracle Solaris オペレーティングシステムの単一のインスタンスを複数のゾーンで共有しているにもかかわらず、アプリケーションコンポーネントを互いに隔離できます。ゾーンは、リソース管理コンポーネントを使用して、アプリケーションが利用可能なシステムリソースをどのように使用するかを制御します。リソース管理機能の詳細は、『[Oracle Solaris 11.2 でのリソースの管理](#)』を参照してください。

ゾーンは、CPU などのリソースの消費量に制限を設けます。ゾーン内で実行されるアプリケーションの処理要件の変化に応じて、これらの制限を拡張することもできます。

追加の隔離として、不変ゾーンと呼ばれる、読み取り専用ルートを保持するゾーンを構成できます。

Oracle Solaris 10 ゾーン

Oracle Solaris 10 ゾーンは、`solaris10` ブランド非大域ゾーンとも呼ばれ、BrandZ テクノロジを使用して Oracle Solaris 11 オペレーティングシステム上で Oracle Solaris 10 アプリケーションを実行します。アプリケーションは、非大域ゾーンによって提供される安全な環境内で、変更されることなく実行されます。これにより、Oracle Solaris 10 システムを使用してアプリケーションの開発、テスト、および配備を行うことができます。これらのブランドゾーン内で実行される作業負荷は、カーネルに加えられた機能拡張を利用して、Oracle Solaris 11 リリース上でのみ利用できる一部の革新的技術を使用できます。これらのゾーンは、Oracle Solaris 10 システムを Solaris 11 上のゾーンに移行するために使用します。`solaris10` ブランドゾーンを NFS サーバーにすることはできません。

詳細は、『[Oracle Solaris 10 ゾーンの実行と使用](#)』を参照してください。

ゾーンブランドの概要

solaris-kz ゾーンと solaris および solaris10 ブランドゾーンの相違点を次に示します。

表 1-1 Oracle Solaris ゾーンブランド機能の比較

| コンポーネント | solaris-kz ブランド | solaris および solaris10 ブランド |
|-----------------|---|--|
| サポートするハードウェア | 指定されたハードウェアでサポートされています。OTN サイトへのリンクまたは HCL を参照してください。 | すべての Oracle Solaris 11.2 システムでサポートされています。HCL を参照してください。 |
| メモリー管理 | 一定量の物理 RAM を solaris-kz 仮想プラットフォームに割り当てる必要があります。 | 大域ゾーンに割り当てられた物理 RAM を共有できます。 |
| カーネルのバージョン | カーネルゾーンは、ホストとは異なるカーネルバージョンまたは SRU レベルを実行できます。 | カーネルバージョンは、大域ゾーンのバージョンと同じである必要があります。 |
| ストレージおよびデバイスの管理 | すべてのストレージアクセスを実行します。カーネルゾーンでは zpool または rootzpool リソースはサポートされません。 | fs、zpool、および dataset zonecfg リソースを使用すると、ファイルシステムレベルでストレージを使用可能にできます。 |
| ネットワーク処理 | 排他的 IP ゾーンのみがサポートされます。 | 排他的 IP ゾーンと共有 IP ゾーンがサポートされます。 |

Oracle Solaris Trusted Extensions システム上での Oracle Solaris ゾーンの使用

Oracle Solaris Trusted Extensions システム上でゾーンを使用する方法については、『[Trusted Extensions 構成と管理](#)』の第 13 章「[Trusted Extensions でのゾーンの管理](#)」を参照してください。Oracle Solaris Trusted Extensions システムでは labeled ブランドしかブートできないことに注意してください。

Oracle Solaris クラスタのゾーンクラスタ

ゾーンクラスタは、Oracle Solaris クラスタソフトウェアの 1 つの機能です。ゾーンクラスタとは、ゾーンクラスタのノードとして機能する非大域ゾーンのグループです。1 つの非大域ゾーンはゾーンクラスタで構成される各グローバルクラスタノード上に作成されます。ゾーンクラスタのノードは solaris ブランドまたは solaris10 ブランドのいずれかにし、クラスタ属性を使用できます。その他のブランドタイプは許可されません。ゾーンによって提供される隔離を使用して、グ

ローバルクラスタ上と同様に、サポートされるサービスをゾーンクラスタ上で実行できます。詳細は、『[Oracle Solaris Cluster システム管理](#)』を参照してください。

ブランドゾーンについて

システム上の非大域ゾーンは、デフォルトでは大域ゾーンと同じオペレーティングシステムソフトウェアを実行します。Oracle Solaris オペレーティングシステムのブランドゾーン (BrandZ) 機能は、Oracle Solaris ゾーンの単純な拡張です。BrandZ フレームワークは、大域ゾーンのオペレーティング環境とは異なるオペレーティング環境を含む非大域ブランドゾーンを作成するために使用します。ブランドゾーンは、Oracle Solaris オペレーティングシステムでアプリケーションを実行するために使用します。BrandZ フレームワークは、さまざまな方法で Oracle Solaris ゾーンインフラストラクチャーを拡張します。これらの拡張には、ゾーン内で異なるオペレーティングシステム環境を実行するための機能を提供するような複雑なものもあれば、新しい機能を提供するために基本のゾーンコマンドを拡張するような簡単なものもあります。たとえば、Oracle Solaris 10 ゾーンは、Oracle Solaris 10 オペレーティングシステムをエミュレートできるブランド非大域ゾーンです。大域ゾーンと同じオペレーティングシステムを共有するデフォルトのゾーンの場合も、*brand* を使用して構成します。

ブランドは、ゾーンにインストールできるオペレーティング環境を定義し、ゾーンにインストールされたソフトウェアが正しく機能するようにゾーン内でのシステムの動作方法を決定します。また、ゾーンのブランドにより、アプリケーションの起動時に正しいアプリケーションタイプが識別されます。すべてのブランドゾーンの管理は、標準のゾーン構造に対する拡張を通して実行されます。管理手順のほとんどはすべてのゾーンで同一です。

定義済みのファイルシステムや特権などの、デフォルトで構成に含まれているリソースについては、ブランドに関するドキュメントで説明されています。

BrandZ はゾーンのツールを次のように拡張します。

- `zonecfg` コマンドを使用して、ゾーンの構成時にゾーンのブランドタイプを設定します。
- `zoneadm` コマンドを使用して、ゾーンのブランドタイプの報告とゾーンの管理を行います。

ラベルが有効にされている Oracle Solaris Trusted Extensions システムにはブランドゾーンを構成およびインストールできますが、ブートされるブランドが認証されたシステム構成上のラベル付きブランドでないかぎり、このシステム構成ではブランドゾーンをブートできません。

構成済み状態にあるゾーンのブランドは変更することができます。ブランドゾーンのインストールが完了したあとは、そのブランドの変更や削除を行うことはできません。



注意 - 既存の Oracle Solaris 10 システムを Oracle Solaris 11 リリースが稼働するシステム上の solaris10 ブランドゾーンに移行することを計画している場合は、最初に既存のゾーンをすべてターゲットシステムに移行する必要があります。solaris10 ゾーンは入れ子にならないため、システムの移行処理は既存のゾーンをすべて使用不可にします。詳細は、『Oracle Solaris 10 ゾーンの作成と使用』の第 3 章「Oracle Solaris 10 ゾーンへの Oracle Solaris 10 ネイティブ非大域ゾーンの移行」を参照してください。

ブランドゾーンで実行中のプロセス

ブランドゾーンでは、ブランドゾーンで実行中のプロセスだけに適用される一連の介入ポイントがカーネル内に用意されています。

- これらのポイントは、syscall パス、プロセスローディングパス、スレッド作成パスなどのパス内に見つかります。
- これらの各ポイントで、ブランドは Oracle Solaris の標準的な動作を補完したり置き換えたりすることができます。

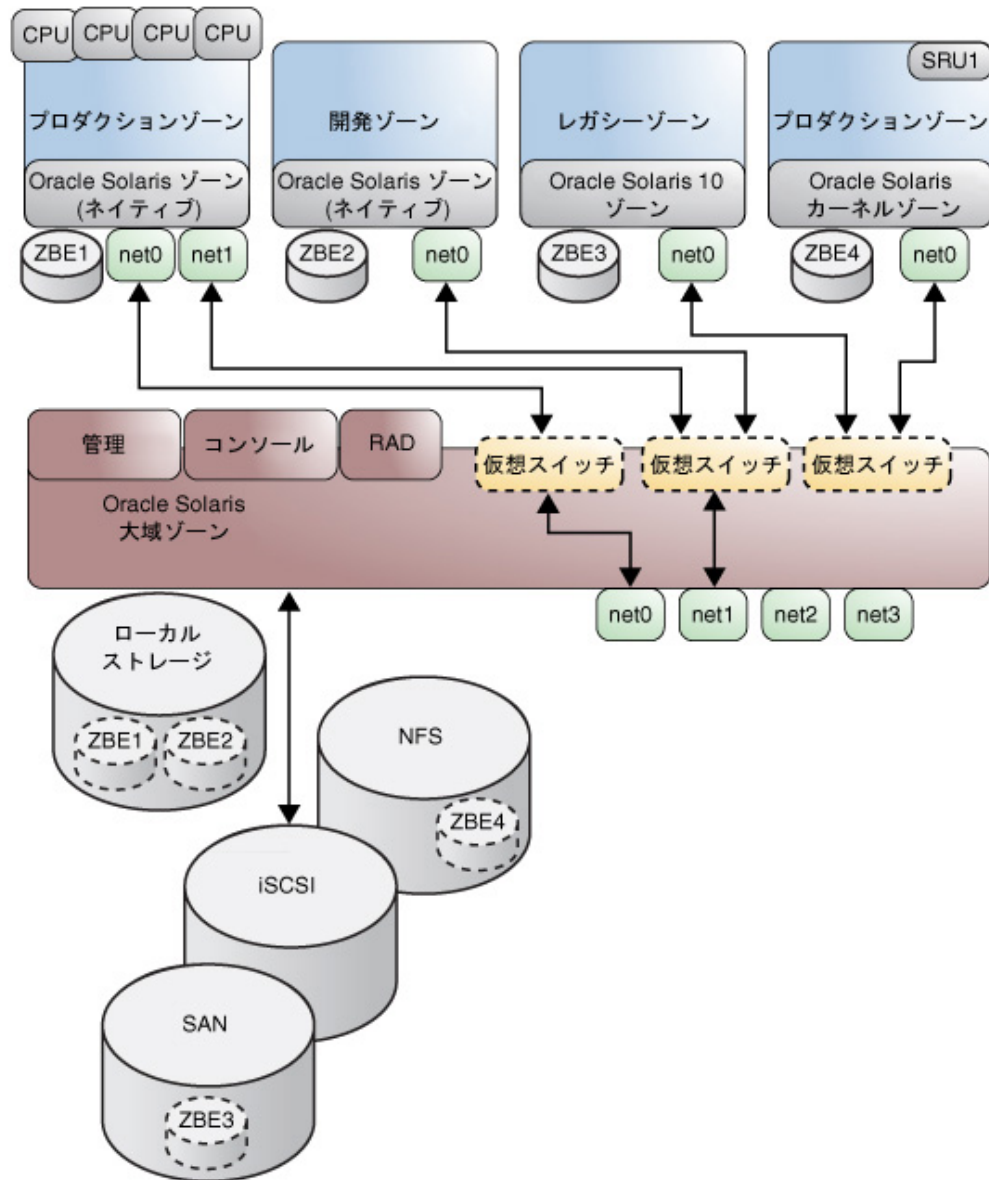
ブランドは librtld_db のプラグインライブラリを提供することもできます。デバッガ ([mdb\(1\)](#) に記載) や DTrace ([dtrace\(1M\)](#) に記載) などの Oracle Solaris のツールは、このプラグインライブラリを使用することによって、ブランドゾーン内で実行しているプロセスのシンボル情報にアクセスできます。

ゾーンでは、静的にリンクされたバイナリはサポートされません。

ゾーンを使用する場合

ゾーンは、多くのアプリケーションを 1 台のサーバー上で統合できる環境で使用するとともに高い効果を発揮します。多数のマシンの管理は複雑でコストがかかるため、より大規模で拡張性の高いサーバーにアプリケーションを統合することが望まれます。

図 1-1 ゾーンによるサーバー統合の例



ゾーンを使用すると、システムのリソースをより効率的に利用できます。リソースの動的再割り当てにより、使用されていないリソースを必要に応じてほかのゾーンに移動できます。障害および

セキュリティーの隔離により、動作状態の悪いアプリケーションのために使用効率の低い専用のシステムを用意する必要がなくなります。ゾーンを使用すると、このようなアプリケーションをほかのアプリケーションと統合できます。

ゾーンを使用すると、総合的なシステムセキュリティーを維持しながら、管理機能の一部を委譲できます。

ゾーンのしくみ

非大域ゾーンは、1つの箱と考えることができます。この箱の中では、システムのほかの部分と相互に作用することなく、1つ以上のアプリケーションを実行できます。ゾーンは、ソフトウェアで定義される柔軟な境界を使用して、ソフトウェアアプリケーションやサービスを隔離します。これにより、Oracle Solaris オペレーティングシステムの同じ1つのインスタンス内で実行される複数のアプリケーションを互いに独立して管理できます。したがって、同じアプリケーションのさまざまなバージョンをそれぞれ異なるゾーンで実行でき、構成の要件を満たすことができます。

ゾーンに割り当てられたプロセスは、同じゾーンに割り当てられたほかのプロセスを操作、モニターしたり、これらのプロセスと直接通信したりできます。システムのほかのゾーンに割り当てられたプロセスや、ゾーンに割り当てられていないプロセスに対しては、このような機能は実行できません。異なるゾーンに割り当てられたプロセスどうしでは、ネットワーク API を介した通信のみ可能です。

ゾーンが独自の排他的 IP インスタンスを持っているのか、IP 層の構成と状態を大域ゾーンと共有しているのかに応じて、IP ネットワーク接続を2通りの方法で構成できます。排他的 IP がデフォルトのタイプです。ゾーンの IP タイプの詳細については、[44 ページの「ゾーンネットワークインタフェース」](#)を参照してください。構成情報については、『[Oracle Solaris ゾーンの作成と使用](#)』の「[ゾーンの構成方法](#)」を参照してください。

Oracle Solaris システムごとに1つの大域ゾーンがあります。大域ゾーンは2つの機能を持っています。大域ゾーンは、システムのデフォルトのゾーンであり、システム全体の管理に使用されるゾーンでもあります。非大域ゾーン（単に「ゾーン」と呼ばれる）が、大域管理者またはゾーンのセキュリティープロファイルを持つユーザーによって作成されていない場合、すべてのプロセスが大域ゾーンで実行されます。

非大域ゾーンの構成、インストール、管理、およびアンインストールは、大域ゾーンからのみ行うことができます。システムハードウェアからブートできるのは、大域ゾーンだけです。物理デバイス、共有 IP ゾーンでのルーティング、動的再構成 (DR) といったシステムインフラストラクチャーの管理は、物理システムで実行されている大域ゾーンでのみ行うことができます。大域ゾーンで実

行されるプロセスは、適切な権限が付与されていれば、ほかのゾーンに関連付けられているオブジェクトにもアクセスできます。

場合によっては、非大域ゾーンの特権付きプロセスには許可されていない操作を、大域ゾーンの特権のないプロセスが実行できることもあります。たとえば、大域ゾーンのユーザーは、システムのすべてのプロセスに関する情報を表示できます。この機能がサイトで問題になる場合は、大域ゾーンへのアクセスを制限します。

大域ゾーンも含め、各ゾーンにはゾーン名が割り当てられます。大域ゾーンの名前は常に `global` となります。各ゾーンには、一意の数値 ID も与えられます。これは、ゾーンのブート時にシステムによって割り当てられます。大域ゾーンは常に ID 0 にマップされます。`zlogin` をカーネルゾーンに対して実行した場合、これは仮想大域ゾーンであるため、ID 0 を持つことも報告されます。ゾーン名と数値 ID については、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」を参照してください。

各ゾーンには、ノード名も割り当てられます。これは、ゾーン名とは完全に独立した名前です。ノード名は、ゾーンの管理者によって割り当てられます。詳細は、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンのノード名」を参照してください。

各ゾーンには、ルートディレクトリのパスが設定されます。これは、大域ゾーンのルートディレクトリに対する相対パスです。詳細は、58 ページの「`zonecfg` コマンドの使用」を参照してください。

デフォルトでは、非大域ゾーンのスケジューリングクラスは、システムのスケジューリングクラスと同じに設定されます。ゾーンのスケジューリングクラスを設定する方法については、39 ページの「スケジューリングクラス」を参照してください。

ブロック型デバイスのマルチパスは、`scsi_vhci(7D)` によって処理されます。構成用に選択する `lu: storage URI` の形式によって、構成の使用方法が決まります。マルチパスでの `lu: URI` の使用の詳細は、`suri(5)` のマニュアルページを参照してください。

ゾーンの機能別のサマリー

次の表に、大域ゾーンと非大域ゾーンの特性をまとめます。

| ゾーンの種類 | 特性 |
|--------|----|
| 大域 | |

| ゾーンの種類 | 特性 |
|--------|--|
| | <ul style="list-style-type: none"> ■ システムによって ID 0 が割り当てられます ■ システムでブートでき、実行される Oracle Solaris カーネルの単一のインスタンスを提供します ■ Oracle Solaris システムソフトウェアパッケージの完全なインストールが含まれています ■ 追加のソフトウェアパッケージや、パッケージを通してインストールされない追加のソフトウェア、ディレクトリ、ファイル、その他のデータが含まれている場合もあります ■ 大域ゾーンにインストールされているすべてのソフトウェアコンポーネントに関する情報を格納した、一貫性のある完全な製品データベースを提供します ■ 大域ゾーンのホスト名やファイルシステムテーブルなど、大域ゾーンのものに固有の構成情報を保持します ■ すべてのデバイスとすべてのファイルシステムが認識される、唯一のゾーンです ■ 非大域ゾーンの存在と構成が認識される、唯一のゾーンです ■ 非大域ゾーンの構成、インストール、管理、またはアンインストールを行うことができる、唯一のゾーンです |
| 非大域 | <ul style="list-style-type: none"> ■ ゾーンのブート時にシステムによってゾーン ID が割り当てられます ■ 大域ゾーンからブートされる Oracle Solaris カーネルの下で処理を共有します ■ 完全な Oracle Solaris オペレーティングシステムソフトウェアパッケージのインストール済みのサブセットが含まれています ■ 追加のインストール済みソフトウェアパッケージを含めることができます ■ 追加のソフトウェア、ディレクトリ、ファイル、およびパッケージによってインストールされない、非大域ゾーンで作成されたその他のデータを含めることができます ■ ゾーンにインストールされているすべてのソフトウェアコンポーネントに関する情報を格納した、一貫性のある完全な製品データベースを保持します ■ ほかのゾーンの存在を認識できません ■ 自身を含め、ゾーンのインストール、管理、アンインストールを行うことはできません ■ 非大域ゾーンのホスト名やファイルシステムテーブルなど、その非大域ゾーンのものに固有の構成情報を保持します ■ 独自のタイムゾーン設定を持つことができます |

非大域ゾーンの管理のしくみ

大域管理者は、スーパーユーザー特権または同等の管理権限を持ちます。大域ゾーンにログインすると、大域管理者はシステム全体をモニターしたり制御したりできます。

非大域ゾーンは「ゾーン管理者」が管理できます。大域管理者は、[36 ページの「admin リソース」](#)で説明されているように、ゾーン管理者に必要な承認を割り当てます。ゾーン管理者の特権は、特定の非大域ゾーンのものに制限されます。

非大域ゾーンの作成のしくみ

非大域ゾーンの構成とインストールは、自動インストール (AI) のクライアントインストールの一環として指定できます。詳細については、『Oracle Solaris 11.2 システムのインストール』を参照してください。Oracle Solaris カーネルゾーンは主に、直接インストール方法を使用して作成されます。カーネルゾーンの作成方法は、『Oracle Solaris カーネルゾーンの作成と使用』の「[カーネルゾーンのインストール](#)」に記載されています。

Oracle Solaris システム上にゾーンを作成する場合、大域管理者は `zonecfg` コマンドを使用して、ゾーンの仮想プラットフォームとアプリケーション環境に対して各種パラメータを指定することによってゾーンを構成します。次に、大域管理者がゾーンをインストールします。大域管理者は、ゾーン管理コマンド `zoneadm` を使用して、ゾーンに対応するファイルシステム階層にソフトウェアをパッケージレベルでインストールします。ゾーンをブートするには、`zoneadm` コマンドを使用します。次に、大域管理者または承認されたユーザーは、`zlogin` コマンドを使用してインストールされたゾーンにログインできます。役割によるアクセス制御 (RBAC) を使用中の場合、ゾーン管理者は `solaris.zone.manage/zonename` の承認を持っている必要があります。

ゾーンの構成については、[第2章「非大域ゾーンの構成の概要」](#)を参照してください。ゾーンのインストールについては、『Oracle Solaris ゾーンの使用』の[第2章「非大域ゾーンのインストール、停止処理、停止、アンインストール、クローニングについて」](#)を参照してください。ゾーンのログインについては、『Oracle Solaris ゾーンの使用』の[第4章「非大域ゾーンへのログインについて」](#)を参照してください。

Oracle Solaris カーネルゾーンを構成およびインストールするには、『Oracle Solaris カーネルゾーンの使用』を参照してください。

非大域ゾーンの状態モデル

非大域ゾーンの状態は、次の7つのいずれかになります。

| | |
|------|--|
| 構成済み | ゾーンの構成は完了し、安定した記憶領域に確定されています。ただし、ゾーンのアプリケーション環境の要素のうち、最初のブート後に指定する必要があるものは、まだ含まれていません。 |
| 不完全 | インストール処理やアンインストール処理の途中は、 <code>zoneadm</code> によってターゲットゾーンの状態が「不完全」に設定されます。処理が正常に完了すると、適切な状態に設定されます。 |

損傷を受けたインストール済みゾーンには、`zoneadm` の `mark` サブコマンドを使用して、不完全のマークを付けることができます。不完全な状態のゾーンは、`zoneadm list -iv` の出力に表示されます。

| | |
|------------|---|
| 使用不可 | <p>ゾーンがインストールされているが、ゾーンを検証、準備、ブート、接続、または移動できないことを示します。ゾーンは次の時点で使用不可状態になります。</p> <ul style="list-style-type: none">■ ゾーンストレージが使用できず、<code>svc:/system/zones:default</code> が起動したとき (システムのブート中など)■ ゾーンストレージが使用できないとき■ アーカイブの抽出に成功したあとでアーカイブベースのインストールが失敗したとき■ ゾーンソフトウェアと大域ゾーンのソフトウェアに互換性がないとき (不適切な <code>-f</code> (強制的) 接続のあとなど) |
| インストール済み | <p>ゾーンの構成はシステム上でインスタンス化されています。<code>zoneadm</code> コマンドは、指定した Oracle Solaris システム上で、この構成を正常に使用できることを確認するために使用します。パッケージはゾーンのルートパスにインストールされます。この状態では、ゾーンに関連付けられた仮想プラットフォームはありません。</p> |
| 準備完了 | <p>ゾーンの仮想プラットフォームが確立されています。カーネルにより <code>zsched</code> プロセスが作成され、ネットワークインタフェースが設定されてゾーンで使用可能になり、ファイルシステムがマウントされ、デバイスの構成が完了しています。システムにより、一意のゾーン ID が割り当てられます。この状態では、ゾーンに関連付けられたプロセスは起動されていません。</p> |
| 稼働 | <p>ゾーンのアプリケーション環境に関連付けられたユーザープロセスが稼働状態です。アプリケーション環境に関連付けられた最初のユーザープロセス (<code>init</code>) が作成されるとすぐに、ゾーンの状態は「稼働」になります。</p> |
| 停止処理中および停止 | <p>これらは、ゾーンの停止処理の間に見られる遷移状態です。ただし、なんらかの理由でゾーンを停止処理できない場合は、ゾーンがどちらかの状態で停止します。</p> |

`zoneadm` コマンドを使用してこれらの状態間の遷移を開始する方法は、『Oracle Solaris ゾーンの作成と使用』の第 3 章「非大域ゾーンのインストール、ブート、停止処理、停止、アンインストール、およびクローニング」および `zoneadm(1M)` のマニュアルページに記載されています。

さらに、Oracle Solaris カーネルゾーンには、現在のゾーン状態に関する追加情報とともにホストに通知するために使用される、3 つの補助状態があります。

| | |
|--------|--|
| 中断中 | プライマリ状態が停止で、補助状態が中断中です。 |
| デバッグ | ゾーンは実行中ですが、ゾーンはネットワーク処理などの外部イベントに 応答できません。zlogin はこの状態を確認して、状態がクリアされるま で待機してから zlogin セッションを開始します。 |
| パニック発生 | ゾーンでパニックが発生しましたが、ゾーンはリポートされるまで外部イベ ントに応答できません。 |

追加情報については、『Oracle Solaris カーネルゾーンの作成と使用』および [solaris-kz\(5\)](#) のマニュアルページを参照してください。

表 1-2 ゾーンの状態に影響を与えるコマンド

| ゾーンの現在の状態 | 適用できるコマンド |
|-----------|---|
| 構成済み | <pre>zonecfg -z zonename verify zonecfg -z zonename commit zonecfg -z zonename delete zoneadm -z zonename attach zoneadm -z zonename verify zoneadm -z zonename install zoneadm -z zonename clone zoneadm -z zonename mark incomplete zoneadm -z zonename mark unavailable</pre> <p>zonecfg コマンドを使用すると、構成済み状態のゾーンの名前を変更できま す。zoneadm コマンドを使用すると、構成済み状態またはインストール済み状態 の Oracle Solaris ゾーンまたは Oracle Solaris 10 ゾーンの名前を変更でき ます。</p> |
| 不完全 | <pre>zoneadm -z zonename uninstall</pre> |
| 使用不可 | <pre>zoneadm -z zonename uninstall</pre> は、指定されたシステムからゾーンをアンイン ストールします。 <pre>zoneadm -z zonename attach</pre> <pre>zonecfg -z zonename</pre> は、zonepath、およびインストール済み状態のときに変更 できないその他のプロパティまたはリソースを変更するために使用できます。 |
| インストール済み | <pre>zoneadm -z zonename ready (オプション) zoneadm -z zonename boot</pre> |

| ゾーンの現在の状態 | 適用できるコマンド |
|-----------|--|
| | <p>zoneadm -z <i>zonename</i> uninstall は、指定されたゾーンの構成をシステムからアンインストールします。</p> <p>zoneadm -z <i>zonename</i> move <i>path</i></p> <p>zoneadm -z <i>zonename</i> detach</p> <p>zonecfg -z <i>zonename</i> を使用すると、プロパティ attr, bootargs, capped-memory, dataset, capped-cpu, dedicated-cpu, device, fs, ip-type, limitpriv, net, rctl, または scheduling-class を追加または削除できます。ゾーンの名前を変更することもできます。</p> <p>zoneadm コマンドを使用すると、構成済み状態またはインストール済み状態の Oracle Solaris ゾーンまたは Oracle Solaris 10 ゾーンの名前を変更できます。</p> <p>zoneadm -z <i>zonename</i> mark <i>incomplete</i></p> <p>zoneadm -z <i>zonename</i> mark <i>unavailable</i></p> |
| 準備完了 | <p>zoneadm -z <i>zonename</i> boot</p> <p>zoneadm halt とシステムリブートを実行すると、準備完了状態のゾーンがインストール済み状態に戻ります。</p> <p>zonecfg -z <i>zonename</i> を使用すると、プロパティの attr, bootargs, capped-memory, dataset, capped-cpu, dedicated-cpu, device, fs, ip-type, limitpriv, net, rctl, または scheduling-class を追加または削除できます。</p> |
| 稼働 | <p>zlogin <i>options</i> <i>zonename</i></p> <p>zoneadm -z <i>zonename</i> reboot</p> <p>zoneadm -z <i>zonename</i> halt を実行すると、準備完了状態のゾーンがインストール済み状態に戻ります。</p> <p>zoneadm halt とシステムのリブートを実行すると、稼働状態のゾーンがインストール済み状態に戻ります。</p> <p>zoneadm -z shutdown は、ゾーンを完全に停止処理します。</p> <p>zonecfg -z <i>zonename</i> を使用すると、プロパティの attr, bootargs, capped-memory, dataset, capped-cpu, dedicated-cpu, device, fs, ip-type, limitpriv, anet, net, rctl, または scheduling-class を追加または削除できません。zonepath リソースは変更できません。</p> |

注記 - zonecfg 経由で変更されたパラメータは、稼働中のゾーンには影響しません。変更を適用するには、ゾーンをリブートする必要があります。

非大域ゾーンの特徴

ゾーンを使用すると、必要に応じてほぼどのような単位にも細かく隔離できます。専用の CPU、物理デバイス、物理メモリーの一部などをゾーンに割り当てる必要はありません。このようなリソースは、1 つのドメインまたはシステムで実行される複数のゾーンに渡って多重化するか、オペレーティングシステムに用意されているリソース管理機能を使ってゾーンごとに割り当てることができます。

ゾーンごとにカスタマイズされたサービスを提供できます。基本的なプロセス隔離を強化するために、同じゾーン内のプロセスのみ互いに認識したりシグナルを送信したりできます。ゾーン間で基本的な通信を行うには、各ゾーンに IP 接続機能を持たせます。あるゾーンで実行中のアプリケーションが、別のゾーンのネットワークトラフィックを監視することはできません。それぞれのパケットストリームが同じ物理インタフェースを通過する場合でも、この隔離は維持されます。

各ゾーンには、ファイルシステム階層の一部が割り当てられます。各ゾーンは、ファイルシステム階層で割り当てられた部分ツリーに限定されます。したがって、特定のゾーンで実行されている作業負荷は、別のゾーンで実行されているほかの作業負荷のディスク上のデータにアクセスすることはできません。

ネームサービスで使用されるファイルは、ゾーン独自のルートファイルシステムのビュー内に置かれます。したがって、異なるゾーンのネームサービスは互いに隔離され、サービスごとに異なる構成を使用できます。

非大域ゾーンでのリソース管理機能の使用

リソース管理機能を使用する場合は、リソース管理制御の境界とゾーンの境界をそろえる必要があります。このように境界をそろえることで、名前空間のアクセス、セキュリティ隔離、およびリソースの使用状況をすべて制御できる、より完成された仮想マシンのモデルを作成できます。

ゾーンで各種のリソース管理機能を使用するための特殊要件については、このドキュメントでこれらの機能に関連する各章を参照してください。

ゾーン関連の SMF サービス

大域ゾーン内のゾーン関連の SMF サービスには、次のようなものがあります。

```
svc:/system/          autoboot=true である各ゾーンを起動します。
zones:default
```

| | |
|--|---|
| <code>svc:/system/zones-install:default</code> | 必要に応じて、初回ブート時にゾーンのインストールを行います。 |
| <code>svc:/application/pkg/zones-proxyd:default</code> | パッケージシステムがゾーンに対してシステムリポジトリへのアクセスを提供するために使用します。 |
| <code>svc:/application/pkg/system-repository:default</code> | ゾーンのインストールやその他の pkg 操作中に使用される pkg データとメタデータをキャッシュするキャッシュプロキシサーバー。 pkg(1) および pkg(5) のマニュアルページを参照してください。 |
| <code>svc:/system/zones-monitoring:default</code> | <code>zonestatd</code> を制御します。 |
| <code>svc:/application/pkg/zones-proxy-client:default</code> | ゾーンプロキシクライアントの SMF サービスは、非大域ゾーンでのみ実行されます。このサービスは、パッケージシステムがゾーンに対してシステムリポジトリへのアクセスを提供するために使用します。 |

非大域ゾーンのモニタリング

現在実行中のゾーンの CPU、メモリ、およびリソース制御の使用率について報告する場合は、『[Oracle Solaris ゾーンの作成と使用](#)』の「[非大域ゾーンでの zonestat ユーティリティの使用](#)」を参照してください。zonestat ユーティリティは、排他的 IP ゾーン内のネットワーク帯域幅の使用効率についても報告を生成します。排他的 IP ゾーンには、独自の IP に関連付けられた状態と 1 つ以上の専用のデータリンクが保持されます。

fsstat ユーティリティを使用して、非大域ゾーンのファイル操作の統計情報を報告できます。fsstat(1M) のマニュアルページおよび『[Oracle Solaris ゾーンの作成と使用](#)』の「[fsstat ユーティリティを使用した非大域ゾーンのモニタリング](#)」を参照してください。

非大域ゾーンによって提供される機能

非大域ゾーンは、次の機能を提供します。

| | |
|--------|--|
| セキュリティ | 大域ゾーン以外のゾーンにプロセスを配置したあとは、そのプロセス自体やそのプロセスの子がゾーンを変更することはできません。 |
|--------|--|

| | |
|----------|---|
| | <p>ネットワークサービスをゾーンで実行できます。ネットワークサービスをゾーンで実行すると、セキュリティ違反が発生した場合の損害を抑えることができます。ゾーン内で実行されているソフトウェアのセキュリティ欠陥を侵入者が悪用できた場合でも、そのゾーン内で可能な一連の操作しか実行できません。ゾーン内で使用できる特権は、システム全体で使用できる特権の一部のみです。</p> |
| 隔離 | <p>複数のアプリケーションが異なる信頼ドメインで動作する場合や、大域リソースへの排他的アクセスを必要とする場合、または、大域の構成を使用すると問題を示すような場合でも、ゾーンを使用することでこれらのアプリケーションを同じマシン上に配備できます。アプリケーションが互いのネットワークトラフィック、ファイルシステムデータ、プロセスの活動などをモニタリングしたり妨害したりすることもできなくなります。</p> |
| ネットワーク隔離 | <p>デフォルトでは、ゾーンは排他的 IP タイプとして構成されます。ゾーンは IP 層で大域ゾーンから分離され、かつ相互に分離されます。この分離は、運用上とセキュリティ上の両方の理由で有用です。ゾーンは、専用の LAN または VLAN を使用して異なるサブネット上で通信する必要があるアプリケーションを統合するために使用できます。ゾーンごとに独自の IP 層のセキュリティ規則を定義することもできます。</p> |
| 仮想化 | <p>ゾーンによって提供される仮想環境では、物理デバイスやシステムのプライマリ IP アドレスとホスト名などの詳細をアプリケーションから隠すことができます。同じアプリケーション環境を、物理的に異なるマシンで維持管理することもできます。仮想化された環境では、各ゾーンを個別に管理できます。非大域ゾーンでゾーン管理者によって行われる操作は、システムのほかの部分に影響を与えません。</p> |
| 隔離単位 | <p>ゾーンを使用すると、ほぼどのような単位にも細かく隔離できます。詳細は、25 ページの「非大域ゾーンの特性」を参照してください。</p> |
| 環境 | <p>セキュリティや隔離の目標を達成するために必要な場合を除き、アプリケーションの実行される環境がゾーンによって変更されることはありません。ゾーンを使用するために、新しい API や ABI にアプリケーションを移植する必要はありません。代わりに、ゾーンでは Oracle Solaris の標準インタフェースとアプリケーション環境が提供されます。ただし、いくつかの制限があります。これらの制限は主に、特権付き操作を実行しようとするアプリケーションに影響を与えます。</p> <p>大域ゾーンで実行されるアプリケーションは、追加のゾーンが構成されたかどうかにかかわらず、変更なしで実行できます。</p> |

このリリースの Oracle Solaris ゾーンについて

このセクションでは、Oracle Solaris カーネルゾーンを含む Oracle Solaris ゾーン機能の概要について説明します。

このリリースのデフォルトの非大域ゾーンは `solaris` です。これについては、このガイドおよび `solaris(5)` のマニュアルページで説明されています。

Oracle Solaris リリースとマシンアーキテクチャーを確認するには、次のように入力します。

```
#uname -r -m
```

[virtinfo\(1M\)](#) のマニュアルページで説明されている `virtinfo` コマンドは、次の情報を取得するために使用します。

- Oracle Solaris 仮想化テクノロジーのシステムサポートを決定します
- Oracle VM Server for SPARC など、Oracle Solaris が実行されている仮想環境のタイプを検出します

`solaris` ゾーンは、ブランドゾーンフレームワーク ([brands\(5\)](#) マニュアルページに記載) を使用して、大域ゾーンにインストールされているソフトウェアと同じソフトウェアがインストールされたゾーンを実行します。`solaris` ブランド非大域ゾーンを使用しているときは、システムソフトウェアは大域ゾーンとの間で同期が常に取れている必要があります。ゾーン内のシステムソフトウェアパッケージは、イメージパッケージングシステム (IPS) を使用して管理されます。IPS は Oracle Solaris 11 リリース上のパッケージングシステムで、`solaris` ゾーンはこのモデルを使用します。

Oracle Solaris 11 Express リリース上に作成されるデフォルトの `ipkg` ゾーンは、`solaris` ゾーンにマップされます。[12 ページの「ipkg ゾーンの solaris ゾーンへの変換について」](#)を参照してください。

自動インストール (AI) のマニフェスト内に指定された各非大域ゾーンは、クライアントのインストールの一部としてインストールおよび構成されます。非大域ゾーンは、大域ゾーンのインストール後の初回リブート時にインストールされ、構成されます。システムの最初のブート時に、ゾーンの自己アセンブリ SMF サービスの `svc:/system/zones-install:default` は、大域ゾーンの AI マニフェスト内で定義された各非大域ゾーンを構成およびインストールします。詳細は、『[Oracle Solaris 11.2 ソフトウェアの追加と更新](#)』を参照してください。また、インストール済みの Oracle Solaris システムでゾーンを手動で構成およびインストールすることも可能です。

パッケージ更新の場合は、`--proxy` オプションを使用してイメージに永続的なプロキシを設定するようにしてください。永続的なイメージプロキシ構成を使用しない場合は、`http_proxy` および `https_proxy` 環境変数を設定できます。

ゾーンは、順次ではなく並列的に更新されるように構成できます。並列更新により、システム上のすべてのゾーンを更新するのに必要な時間が大幅に短縮されます。

ゾーンは、デフォルトでは排他的 IP タイプで作成されます。ネットワーク接続構成が指定されない場合、VNIC は `anet` リソースによってゾーン構成内に自動的に含まれます。詳細は、[44 ページの「ゾーンネットワークインタフェース」](#)を参照してください。

ゾーンの `mac-address` を取得するために使用される `auto-mac-address` については、[68 ページの「リソースタイプのプロパティ」](#)のエントリ `anet` を参照してください。

共有ストレージ上の `solaris` ゾーンには、`zonecfg rootzpool` リソースがあります。ゾーンは専用の `zpool` にカプセル化されます。共有ストレージ上のゾーンは、ゾーン用の共有ストレージリソースにアクセスし、それらを管理します。カーネルゾーンには、`zpool` または `rootzpool` リソースはありません。`solaris` ブランドゾーンは、ゾーン `device` リソース、`zpool` リソース、および `rootzpool` リソースに次の共有ストレージを使用できます。

- iSCSI
- FC LUN
- DAS

IPoIB (IP over InfiniBand) データリンクを指定するために使用されるプロパティは、`zonecfg anet` リソースに使用できます。IPoIB は、`solaris` と `solaris10` の両方のブランドゾーンでサポートされています。

RDS (Reliable Datagram Sockets) IPC プロトコルは、排他的 IP 非大域ゾーンと共有 IP 非大域ゾーンの両方でサポートされています。

`fsstat` ユーティリティーは、ゾーンをサポートするように拡張されました。`fsstat` ユーティリティーは、ゾーン別および集計の統計情報を提供します。

『Oracle Solaris ゾーンの実行と使用』の「ゾーンでの NFS サーバーの実行」のセクションで説明されているように、`solaris` ゾーンは NFS サーバーにできます。

試行 (「ドライラン」とも呼ばれる) の `zoneadm attach -n` は、`zonecfg` の検証を提供しますが、パッケージ内容の検証は実行しません。

ファイルを引数に取る `zoneadm` のすべてのオプションには、絶対パスが必要です。

Oracle Solaris 10 ゾーンは、Oracle Solaris 11 上に Oracle Solaris 10 環境を提供します。Oracle Solaris 11 システム上の `solaris10` ゾーンに Oracle Solaris 10 システムまたはゾーンを移行できます。『[Oracle Solaris 10 ゾーンの作成と使用](#)』を参照してください。

`zonep2vchk` ツールは、Oracle Solaris 11 システムまたは Oracle Solaris 10 システムの Oracle Solaris 11 リリースを実行しているシステム上のゾーンへの移行に影響を与える可能性のある、ネットワークの問題を含む問題を特定します。`zonep2vchk` ツールは、移行が始まる前にソースシステム上で実行されます。このツールは、ターゲットシステム上で使用するための `zonecfg` スクリプトも出力します。このスクリプトは、ソースシステムの構成と一致するゾーンを作成します。詳細は、『[Oracle Solaris ゾーンの作成と使用](#)』の第 7 章「[ゾーンの移行と zonep2vchk ツールについて](#)」を参照してください。

Oracle Solaris 10 リリース上の `solaris` ゾーンと `native` ゾーンとの間の次の相違に留意してください。

- Oracle Solaris 11 システムでは、Oracle Solaris 10 システム上のデフォルトである `native` ブランドではなく、`solaris` ブランドが作成されます。
 - `solaris` ゾーンは、完全ルートタイプのみです。
- Oracle Solaris 10 上で使用できるネイティブゾーンの疎ルートタイプは、SVR4 パッケージ管理システムを使用し、IPS はこのシステムを使用しません。疎ルートタイプと同様の読み取り専用ルートゾーン構成を使用できます。
- このリリースのゾーンには、これらの領域での Oracle Solaris 10 リリースとは異なる、次のソフトウェア管理関連の機能があります。
 - IPS パッケージと SVR4 パッケージ。
 - インストール、切り離し、接続、および Physical-To-Virtual 機能。
 - 非大域ゾーンルートが ZFS™ データセットである。

大域ゾーンにインストールされたパッケージは、現在および将来のすべてのゾーンにインストールされなくなります。一般的に、IPS パッケージと SVR4 パッケージの両方において、大域ゾーンのパッケージ内容は、各ゾーンのパッケージ内容に影響しません。

- 非大域ゾーンはブート環境を使用します。ゾーンは、ZFS ブート環境 (BE) を管理するためのユーザーインタフェースコマンドである `beadm` と統合されています。

`beadm` コマンドは、大域ゾーン内と同様に、`pkg update` 用にゾーンの内部でサポートされます。`beadm` コマンドは、ゾーンに関連付けられている、アクティブでないゾーン BE を削除できます。[beadm\(1M\)](#) のマニュアルページを参照してください。

- 有効になっているすべての IPS パッケージリポジトリが、ゾーンのインストール中にアクセス可能である必要があります。詳細は、『Oracle Solaris ゾーンの作成と使用』の「構成済みのゾーンをインストールする方法」を参照してください。
- ゾーンソフトウェアは、最初は最小限に抑えられています。ゾーンが必要とするその他のパッケージは、追加する必要があります。詳細は、『Oracle Solaris 11.2 ソフトウェアの追加と更新』を参照してください。

ゾーンでは、次のような Oracle Solaris の各製品と機能を使用できます。

- Oracle Solaris ZFS 暗号化
- ネットワーク仮想化および QoS
- CIFS と NFS

solaris-kz ブランドゾーンでは、次の機能を構成できません。

- FC サービス
- FCoE サービス

ライブゾーン再構成

リポートせずに実行中の solaris ゾーンまたは solaris10 ゾーンのライブ構成を再構成または報告するには、ライブゾーン再構成を使用します。変更は、一時的または永続的に行うことができます。

solaris-kz ブランドゾーンのライブ構成情報を報告するには、ライブゾーン再構成を使用します。

詳細は、『Oracle Solaris ゾーンの作成と使用』を参照してください。

◆◆◆ 第 2 章

非大域ゾーンの構成の概要

この章では、非大域ゾーンの構成の概要について説明します。

この章の内容は次のとおりです。

- 33 ページの「ゾーンのリソースについて」
- 35 ページの「インストール前の構成処理」
- 35 ページの「ゾーンのコンポーネント」
- 58 ページの「zonecfg コマンドの使用」
- 59 ページの「zonecfg のモード」
- 62 ページの「ゾーン構成データ」
- 78 ページの「Tecla コマンド行編集ライブラリ」

ゾーンの構成について学んだあとで、『Oracle Solaris ゾーンの作成と使用』の第 1 章「非大域ゾーンの計画および構成方法」に進み、システムにインストールする非大域ゾーンを構成します。

ゾーンのリソースについて

ゾーン内で制御できるリソースには、次のものがあります。

- リソースプールまたは割り当てられる CPU。マシンリソースの区分に使用されます。
- リソース制御機能。システムリソースに対する制約メカニズムを提供します。
- スケジューリングクラス。使用可能な CPU リソースのゾーン間での割り当てを、それらの重要性に基づいて制御できます。この重要性は、各ゾーンに割り当てる CPU リソースの「配分」で表します。

ゾーン管理での権利プロファイルと役割の使用

プロファイルと役割については、『Oracle Solaris 11 セキュリティーガイドライン』の「アプリケーションの保護と分離」を参照してください。

zonecfg template プロパティ

次の場合にプロパティを変更するかどうか、およびどのように変更するかを定義するには、zonecfg template プロパティを使用します。

- 新しいリソースインスタンスが構成に追加されたとき。
- 構成のクローニング中に、いくつかのプロパティに一意の値が必要なとき。template プロパティでトークンを使用して、これらの一意の値を指定します。

表 2-1 zonecfg template トークン

| トークン | 説明 | 使用法 |
|--------------|--|---|
| %zonename | ゾーンの名前。 | ブランドのメタデータ内および zonecfg 内で、ユーザーまたはテンプレート値からの入力として使用できます。 |
| %network-id | ネットワークリソース net および anet の一意のインスタンス番号。この番号は、大域ゾーン内の net および anet リソースで一意です。 | ブランドのメタデータ内で、id プロパティの net および anet リソースのデフォルトの属性として使用できます。 |
| %resource-id | net および anet を除くすべてのリソースについての、大域ゾーン内の特定のリソースの大域有効範囲での一意のインスタンス番号。 | ブランドのメタデータ内で、id プロパティのデフォルト属性として使用できます。 |
| %id | リソースの id プロパティ値である一意のインスタンス番号。 | zonecfg 内で、ユーザーまたはテンプレート値からの入力として使用できます。id プロパティをサポートするリソース有効範囲内で使用する必要があります。 |
| %% | % に評価されます。 | ブランドのメタデータ内および zonecfg 内で、ユーザー |

| トークン | 説明 | 使用法 |
|------|----|------------------|
| | | からの入力として使用できません。 |

ゾーンのリモート管理デーモン (RAD) のモジュール構成は、プロパティテンプレートを使用することで、変更を表現、適用、または実装する体系的な方法を提供します。zonemgr(3RAD) のマニュアルページを参照してください。最初に rad-zonemgr パッケージがシステムにインストールされておらず、あとで pkg install を使用してインストールした場合、rad:local を再起動する必要があります。また、rad:remote が実行されていた場合は、これも再起動します。再起動するには、svcadm(1M) を使用します。RAD デーモンがモジュールをロードしたことを確認します。

インストール前の構成処理

非大域ゾーンをインストールしてシステムで使用する前に、ゾーンを構成する必要があります。

zonecfg コマンドを使用すると、構成を作成したり、指定されたリソースやプロパティが仮定のシステム上で有効かどうかを判定したりできます。特定の構成について zonecfg で実行される検査では、次のことが確認されます。

- ゾーンパスが指定されていること。
- 各リソースの必須プロパティがすべて指定されていること。
- 構成に矛盾がないこと。たとえば、anet リソースが存在する場合、ゾーンは排他的 IP タイプであり、共有 IP ゾーンにはできません。また、別名が付いたデータセットでデバイスに関する潜在的競合が存在する場合、zonecfg コマンドは警告を出力します。

zonecfg コマンドの詳細は、zonecfg(1M) のマニュアルページを参照してください。

ゾーンのコンポーネント

このセクションでは、構成できる必須および省略可能なゾーンコンポーネントについて説明します。ゾーン名とゾーンのパスだけが必要です。詳細は、62 ページの「ゾーン構成データ」を参照してください。

ゾーンの名前とパス

ゾーンの名前を選択する必要があります。パスを指定しない場合、`zonepath` のデフォルト値は `/system/zones/zonename` です。

ゾーンの名前とパスを選択する場合、ゾーンは ZFS データセット上に存在する必要があります。ZFS データセットは、ゾーンのインストール時または接続時に自動的に作成されます。ZFS データセットを作成できない場合、ゾーンのインストールや接続は行われません。ゾーンのパスの親ディレクトリは、データセットでもなければならぬことに注意してください。

パスを指定しない場合、`zonepath` のデフォルト値は `/system/zones/zonename` です。

ゾーンの自動ブート

`autoboot` プロパティの設定は、大域ゾーンのブート時にゾーンが自動的にブートされるかどうかを決定します。ゾーンサービス `svc:/system/zones:default` も有効になっている必要があります。

不変ゾーンの `file-mac-profile` プロパティ

`solaris` ゾーンでは、`file-mac-profile` は、読み取り専用ルートで不変ゾーンを構成するために使用します。

詳細は、『[Oracle Solaris ゾーンの実装と使用](#)』の第 12 章「不変ゾーンの構成と管理」を参照してください。

admin リソース

`admin` 設定を使用すると、ゾーン管理の承認を設定できます。承認を定義するための推奨の方法は、`zonecfg` コマンドによる方法です。

| | |
|---------------------------------|--|
| <code>user</code> | ユーザー名を指定します。 |
| <code>auths</code> | ユーザー名に対して承認を指定します。 |
| <code>solaris.zone.login</code> | RBAC を使用中の場合、対話型ログインには、 <code>solaris.zone.login/zonename</code> の承認が必 |

要です。ゾーン内では、パスワード認証が実行されます。

`solaris.zone.manage` RBAC を使用中の場合、非対話型ログイン、またはパスワード承認の省略のために、`solaris.zone.manage/zonename` の承認が必要です。

`solaris.zone.clonefrom` RBAC を使用中の場合、別のゾーンのコピーを作成するサブコマンドは `solaris.zone.clonefrom/source_zone` の承認を必要とします。

承認の詳細は、[auths\(1\)](#)、[auth_attr\(4\)](#)、および [user_attr\(4\)](#) を参照してください。

dedicated-cpu リソース

`dedicated-cpu` リソースは、非大域ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用に割り当てることを指定します。ゾーンのブート時に、ゾーンの実行中に使用される一時プールが動的に作成されます。

`zonecfg` で指定すると、移行時にプールの設定が伝達されます。

`dedicated-cpu` リソースは、`ncpus` の制限を設定し、必要に応じて `importance` も設定します。

`ncpus` CPU の数を指定するか、CPU の数の範囲を 2-4 などと指定します。リソースプールの動的な動作を得るために範囲を指定する場合は、次の手順も実行してください。

- `importance` プロパティを設定します。
- `poold` サービスを有効にします。手順については、『[Oracle Solaris 11.2 でのリソースの管理](#)』の「[svcadm を使用して動的リソースプールサービスを有効にする方法](#)」を参照してください。

`importance` 動的な動作を得るために CPU 範囲を使用する場合は、`importance` プロパティも設定してください。`importance` は「省略可能な」プロパティであり、プールの相対的な重要性を定義します。このプロパティが必要となるのは、`ncpus` に範囲を指定した場合で、`poold` によって管理される動的リソースプールを使用しているときだけです。`poold` が実行されていない場合、`importance` は無視されます。`poold` が実行されている場合、`importance` が設定されていないと、`importance` はデフォルト値の 1 になります。詳細は、『[Oracle Solaris 11.2 でのリソースの管理](#)』の「[pool.importance プロパティの制約](#)」を参照してください。

次のプロパティは、cpus、cores、および sockets に永続的な dedicated-cpu リソースを設定するために使用されます。

| | |
|---------|--------------------------|
| cpus | 特定の CPU を永続的にゾーンに割り当てます。 |
| cores | 特定のコアを永続的にゾーンに割り当てます。 |
| sockets | 指定された数のソケットを永続的に割り当てます。 |

注記 - capped-cpu リソースと dedicated-cpu リソースには互換性がありません。cpu-shares リソース制御と dedicated-cpu リソースには互換性がありません。

注記 - 使用可能な CPU の数に自動的にサイズ設定して拡大縮小するアプリケーションが capped-cpu 制限を認識しないことがあります。すべての CPU が使用可能として認識されると、Oracle データベースや Java 仮想マシン (JVM) などのアプリケーションでスケーリングとパフォーマンスに影響を与える可能性があります。アプリケーションが動作していないか、使用できないように見えることがあります。パフォーマンスが重要である場合は、capped-cpu で JVM を使用しないでください。影響を受けるカテゴリ内のアプリケーションでは dedicated-cpu リソースを使用できます。

solaris-kz のみ: virtual-cpu リソース

カーネルゾーンの CPU の数を設定するには、virtual-cpu リソースを使用します。カーネルゾーン専用のホスト CPU は、ncpus 値によって定義されます。デフォルトのカーネルゾーン構成の CPU は 1 つです。virtual-cpu プロパティを追加すると、さらに CPU をカーネルゾーンに追加できます。

dedicated-cpu リソースがすでに定義されている場合、仮想プラットフォームで構成されている仮想 CPU のデフォルト数は、dedicated-cpu リソース内の ncpus の範囲の下限値と一致します。dedicated-cpu および virtual-cpu の両方のリソースを設定する必要はありません。

capped-cpu リソース

capped-cpu リソースは、1 つのプロジェクトまたは 1 つのゾーンで消費可能な CPU リソース量に対して絶対的できめの細かい制限を設けます。プロセッサセットと組み合わせて使用すると、CPU キャップはセット内の CPU 使用率を制限します。capped-cpu リソースには、小数点第 2 位までの正の小数である 1 つの ncpus プロパティがあります。このプロパティは、CPU のユニット数に対応しています。このリソースには範囲を指定できません。このリソースには小数

を指定できます。ncpus を指定する場合、1 の値は 1 つの CPU の 100% を意味します。1.25 の値は 125% を意味します。100% がシステム上の 1 つの CPU の上限となります。

注記 - capped-cpu リソースと dedicated-cpu リソースには互換性がありません。

注記 - 使用可能な CPU の数に自動サイズ設定して自動的に拡大縮小するアプリケーションが capped-cpu 制限を認識しないことがあります。すべての CPU が使用可能として認識されると、Oracle データベースや Java 仮想マシン (JVM) などのアプリケーションでスケージングとパフォーマンスに影響を与える可能性があります。アプリケーションが動作していないか、使用できないように見えることがあります。パフォーマンスが重要である場合は、capped-cpu で JVM を使用しないでください。影響を受けるカテゴリ内のアプリケーションでは dedicated-cpu リソースを使用できます。[37 ページの「dedicated-cpu リソース」](#)を参照してください。

スケジューリングクラス

公平配分スケジューラ (FSS) を使用すると、使用可能な CPU リソースのゾーン間での割り当てを、ゾーンの重要性に基づいて制御できます。この重要性は、各ゾーンに割り当てる CPU リソースの「配分」で表します。CPU リソースのゾーン間での割り当てを管理するために FSS を使用していない場合でも、ゾーン内のプロジェクトに配分を設定するために FSS を使用するよう、ゾーンのスケジューリングクラスを設定することができます。

cpu-shares プロパティを明示的に設定すると、公平配分スケジューラ (FSS) はそのゾーンのスケジューリングクラスとして使用されます。ただし、この場合に望ましい FSS の使用方法は、dispadmin コマンドを使用して、FSS をシステムのデフォルトのスケジューリングクラスに設定する方法です。このようにすると、すべてのゾーンがシステムの CPU リソースの公平配分を受けることができます。ゾーンに対して cpu-shares が設定されていない場合、そのゾーンはシステムのデフォルトのスケジューリングクラスを使用します。ゾーンのスケジューリングクラスは、次の処理によって設定されます。

- zonecfg の scheduling-class プロパティを使ってゾーンのスケジューリングクラスを設定できます。
- リソースプール機能を使ってゾーンのスケジューリングクラスを設定できます。ゾーンがプールに関連付けられている場合、そのプールの pool.scheduler プロパティに有効なスケジューリングクラスが設定されていれば、ゾーンで実行されるプロセスは、デフォルトでそのスケジューリングクラスで実行されます。『Oracle Solaris 11.2 でのリソースの管理』の「リソースプールの紹介」および『Oracle Solaris 11.2 でのリソースの管理』の「プールをスケジューリングクラスに対応付ける方法」を参照してください。

- `cpu-shares` リソース制御が設定されている場合で、別のアクションを通じて FSS がゾーンのスケジューリングクラスとして設定されていないときは、ゾーンのブート時に `zoneadmd` によってスケジューリングクラスが FSS に設定されます。
- ほかの処理を通してスケジューリングクラスが設定されていない場合、ゾーンはシステムのデフォルトのスケジューリングクラスを継承します。

`priocntl` ([`priocntl\(1\)`](#) のマニュアルページに記載) を使用すると、デフォルトのスケジューリングクラスの変更やリブートを行うことなく、実行中のプロセスを別のスケジューリングクラスに移動できます。

物理メモリーの制御と `capped-memory` リソース

`capped-memory` リソースは、`physical`、`swap`、および `locked` メモリーの制限を設定します。各制限はオプションですが、少なくとも 1 つは設定する必要があります。`capped-memory` リソースを使用するには、`resource-cap` パッケージが大域ゾーン内にインストールされている必要があります。[38 ページの「`capped-cpu` リソース」](#)も参照してください。

- 大域ゾーンから `rcapd` を使用してゾーンのメモリー上限を設定する場合は、このリソースの値を決定します。`capped-memory` リソースの `physical` プロパティは、ゾーンの `max-rss` 値として `rcapd` で使用されます。
- `capped-memory` リソースの `swap` プロパティは、`zone.max-swap` リソース制御を設定するための望ましい方法です。
- `capped-memory` リソースの `locked` プロパティは、`zone.max-locked-memory` リソース制御を設定するための望ましい方法です。

注記 - 通常はアプリケーションが多量のメモリーをロックすることはありませんが、ゾーンのアプリケーションによってメモリーがロックされることがわかっている場合は、ロックされるメモリーを設定するとよいでしょう。ゾーンの信頼が問題になる場合は、ロックされるメモリーの上限を、システムの物理メモリーの 10 パーセントまたはゾーンの物理メモリー上限の 10 パーセントに設定することもできます。

詳細は、『[Oracle Solaris 11.2 でのリソースの管理](#)』の第 10 章「リソース上限デーモンによる物理メモリーの制御について」、『[Oracle Solaris 11.2 でのリソースの管理](#)』の第 11 章「リソース上限デーモンの管理のタスク」、および『[Oracle Solaris ゾーンの作成と使用](#)』の「[ゾーンの構成方法](#)」を参照してください。ゾーンのリソース上限を一時的に設定するには、『[Oracle Solaris 11.2 でのリソースの管理](#)』の「[ゾーンに一時的なリソース上限を指定する方法](#)」を参照してください。

solaris および solaris10 のみ: rootzpool リソース

zonecfg ユーティリティのオプションの rootzpool リソースは、solaris および solaris10 ブランドゾーンのゾーンインストール用に専用の zpool を作成するために使用されます。ゾーンのルート zpool は、1 つ以上の URI (Universal Resource Identifier) によって定義された共有ストレージデバイスでホストできます。必須の storage プロパティは、ゾーンのルート zfs ファイルシステムを格納するストレージオブジェクトの URI を識別します。特定のゾーンに対して定義できる rootzpool は 1 つだけです。このストレージは、ゾーンがブートされたときに、そのゾーンのために自動的に構成されます。

対応する zpool は、ゾーンのインストールまたはゾーンの接続操作中に自動的に作成またはインポートされます。rootzpool と zpool の両方のリソースで、ゾーンがインストールされるとすぐに zpool ミラーを自動的に作成できます。詳細は、『Oracle Solaris ゾーンの実装と使用』の第 14 章「共有ストレージでの Oracle Solaris ゾーンの使用開始」を参照してください。

ゾーンのアンインストールまたは切り離しを行うと、次のアクションが実行されます。

- 対応する zpool が自動的にエクスポートまたは破棄されます。
- ストレージリソースが自動的に構成解除されます。

事前作成済みの zpool をゾーンのインストールに再利用するには、zpool をシステムからエクスポートする必要があります。

ゾーンのフレームワークでは、次の URI タイプがサポートされています。

- dev

ローカルデバイスパスの URI

形式:

`dev:local-path-under/dev`

`dev://absolute-path-with-dev`

`dev:absolute-path-with-dev`

例:

`dev:dsk/c7t0d0s0`

`dev:///dev/dsk/c7t0d0s0`

`dev:/dev/dsk/c7t0d0s0`

`dev:chassis/SYS/HD1/disk`

■ lu (論理ユニット)

ファイバチャネル (FC) および Serial Attached SCSI (SAS)

形式:

lu: *luname.naa.ID*

lu: *luname.eui.ID*

lu: *initiator.naa.ID, target.naa.ID, luname.naa.ID*

lu: *initiator.naa.ID, target.naa.ID, luname.eui.ID*

例:

lu: *luname.naa.5000c5000288fa25*

lu: *luname.eui.0021280001cf80f6*

lu: *initiator.naa.2100001d38089fb0, target.naa.2100001d38089fb0, luname.naa.5000c5000288fa25*

lu: *initiator.naa.2100001d38089fb0, target.naa.2100001d38089fb0, luname.eui.0021280001cf80f6*

■ iscsi

iSCSI の URI

形式:

iscsi://*luname.naa.ID*

iscsi://*luname.eui.ID*

iscsi://*host[:port]/luname.naa.ID*

iscsi://*host[:port]/luname.eui.ID*

iscsi://*target.IQN, lun.LUN*

iscsi://*host[:port]/target.IQN, lun.LUN*

例:

iscsi://*luname.eui.0021280001cf80f6*

iscsi://*luname.naa.600144f03d70c80000004ea57da10001*

iscsi://[:1]/*luname.naa.600144f03d70c80000004ea57da10001*

iscsi://127.0.0.1/*luname.naa.600144f03d70c80000004ea57da10001*

iscsi://*hostname:1234/luname.eui.0021280001cf80f6*

iscsi://*hostname:3260/luname.naa.600144f03d70c80000004ea57da10001*

iscsi://127.0.0.1/*target.iqn.com.sun:02:d0f2d311-f703, lun.0*

iscsi://*target.iqn.com.sun:02:d0f2d311-f703, lun.6*

iscsi://[:1]:*1234/target.iqn.com.sun:02:d0f2d311-f703, lun.2*

```
iscsi://hostname:1234/target.iqn.com.sun:4db41b76-e3d7-cd2f-bf2d-9abef784d76c, lun.0
```

ストレージの URI に基づいて共有オブジェクトを管理するには、`suriadm` ツールを使用します。ID、NAA (Name Address Authority)、および既存のストレージオブジェクトの URI の取得については、[suriadm\(1M\)](#) および [suri\(5\)](#) のマニュアルページを参照してください。

システムは、関連するゾーンに対して新規作成またはインポートされた `rootzpool` に名前を割り当てます。割り当てられる名前は、`zonename_rpool` という形式になります。

`storage` プロパティは、`rootzpool` リソーススコープ内から次のコマンドを使用して管理されます。

- `add storage URI string`
- `remove storage URI string`

zpool リソースの自動追加

`zpool` を非大域ゾーンに委任するには、`zonecfg` ユーティリティーでオプションの `zpool` リソースを構成します。`zpool` は、ゾーンがブートされたときに、そのゾーンのために自動的に構成されます。

対応する `zpool` は、ゾーンのインストールまたはゾーンの接続操作中に自動的に作成またはインポートされます。

ゾーンのアンインストールまたは切り離しを行うと、次のアクションが実行されます。

- 対応する `zpool` が自動的にエクスポートまたは破棄されます。
- ストレージリソースが自動的に構成解除されます。

必須の `storage` プロパティは、このリソースに関連付けられたストレージオブジェクトの URI を識別します。

`storage` プロパティは、`zpool` リソーススコープ内の次の設定を使用して管理されます。

- `add storage URI string`
- `remove storage URI string`

`zpool` リソースには `name` プロパティが必須です。このプロパティは、ゾーンに委任される `zpool` の名前で使用されます。ZFS ファイルシステムの `name` コンポーネントにスラッシュ (`/`) を含めることはできません。

新規作成またはインポートされた zpool に割り当てられる名前は、name プロパティの値です。これは、非大域ゾーン内で表示可能な zpool 名です。新規作成またはインポートされた zpool 名に割り当てられる名前は、大域ゾーンから表示される際には `zonename_name` という形式になります。

注記 - ストレージオブジェクトに既存のパーティション、zpool、または UFS ファイルシステムが含まれている場合は、ゾーンのインストールが失敗することがあります。詳細は、『[Oracle Solaris ゾーンの実装と使用](#)』の「[構成済みのゾーンをインストールする方法](#)」のステップ 4 を参照してください。

ゾーンネットワークインタフェース

ネットワーク接続を提供するために `zonecfg` ユーティリティによって構成されるゾーンネットワークインタフェースは、ゾーンのブート時に自動的に設定されてゾーン内に配置されます。

インターネットプロトコル (IP) 層は、ネットワークのパケットの受信と配信を行います。この層には、IP ルーティング、アドレス解決プロトコル (ARP)、IP セキュリティアーキテクチャー (IPsec)、および IP フィルタが含まれます。

非大域ゾーンに使用できる IP タイプには、共有 IP と排他的 IP の 2 種類があります。排他的 IP がデフォルトの IP タイプです。共有 IP ゾーンは、大域ゾーンとネットワークインタフェースを共有します。大域ゾーン内の構成は、共有 IP ゾーンを使用するために `ipadm` ユーティリティによって実行する必要があります。排他的 IP ゾーンには専用のネットワークインタフェースが必要です。`anet` リソースを使用して排他的 IP ゾーンを構成すると、専用の VNIC が自動的に作成され、そのゾーンに割り当てられます。自動化された `anet` リソースを使用することによって、大域ゾーン内にデータリンクを作成および構成し、非大域ゾーンにデータリンクを割り当てる必要がなくなります。次のことを行うには、`anet` リソースを使用します。

- 大域ゾーンの管理者が、非大域ゾーンに割り当てられたデータリンク用の特定の名前を選択できるようにする
- 複数のゾーンで同じ名前のデータリンクを使用できるようにする

下位互換性を維持するために、事前構成済みのデータリンクを非大域ゾーンに割り当てることができます。

各タイプの IP 機能については、『[Oracle Solaris ゾーンの実装と使用](#)』の「[排他的 IP 非大域ゾーンにおけるネットワーク](#)」および『[Oracle Solaris ゾーンの実装と使用](#)』の「[共有 IP 非大域ゾーンにおけるネットワーク](#)」を参照してください。

注記 - ゾーンを実行するシステムで、『[Oracle Solaris 11.2 でのネットワークのセキュリティ保護](#)』で説明されているリンク保護を使用できます。この機能は、大域ゾーン内で構成されません。

データリンクについて

データリンクとは、OSI プロトコルスタックのレイヤー 2 物理インタフェースのことです。このインタフェースは、システム内で STREAMS DLPI (v2) インタフェースとして表されます。このようなインタフェースは、TCP/IP などのプロトコルスタックの下で接続できます。データリンクは、「物理インタフェース」(たとえば、ネットワークインタフェースカード (NIC)) とも呼ばれます。データリンクは、`zonecfg(1M)` を使用して構成される `physical` プロパティです。`physical` プロパティは `VNIC` でもかまいません。

Oracle Solaris 11 のデフォルトでは、物理ネットワークデバイス名は、`nxge0` などのデバイスドライバ名ではなく、`net0` などの総称名を使用します。

`solaris` ゾーンでの IPoIB (IP over Infiniband) の使用については、[68 ページの「リソースタイプのプロパティ」](#)の `anet` の説明を参照してください。

エラスティック仮想スイッチとゾーンについて

`evs` および `vport` プロパティを設定してエラスティック仮想スイッチ (EVS) に接続する `anet` リソースでは、その `anet` リソースのプロパティは、`evs` と `vport` のペア内にカプセル化されません。EVS `anet` リソースの次のプロパティは変更できません。

- `mac-address`
- `mtu`
- `maxbw`
- `priority`
- `allowed-address`
- `vlan-id`
- `defrouter`
- `lower-link`

EVS `anet` リソースに設定できる唯一のプロパティは次のとおりです。

- `linkname`

- `evs`
- `vport`
- `configure-allowed-address`

`tenant` リソースを設定する必要もあります。テナントは、名前空間の管理に使用されま
す。`tenant` 内で定義されている EVS リソースは、そのテナントの名前空間の外部からは見え
ません。

`evszone` という名前のゾーンの次の入力、`tenantA` という名前のテナントの `tenant` リソース
を設定します。`zonecfg anet` リソースプロパティは、`evsa` という名前の EVS および `vport0` と
いう名前の VPort に接続する `anet` リソースを持つゾーンの VNIC を作成します。

```
zonecfg:evszone> set tenant=tenantA
```

```
zonecfg:evszone> add anet
```

```
zonecfg:evszone> set evs=EVSA
```

```
zonecfg:evszone> set vport=vport0
```

詳細は、『[Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理](#)』の第 5 章
「[エラスティック仮想スイッチについて](#)」を参照してください。

共有 IP 非大域ゾーン

共有 IP ゾーンは、大域ゾーンからの既存の IP インタフェースを使用します。ゾーンには、1 つ
以上の専用の IP アドレスが存在する必要があります。共有 IP ゾーンは、IP 層の構成と状態
を大域ゾーンと共有します。次の両方の条件が満たされる場合、ゾーンは共有 IP インスタンス
を使用すべきです。

- 大域ゾーンと非大域ゾーンが同じサブネット上に存在するかどうかに関係なく、非大域ゾ
ーンが大域ゾーンによって使用されるデータリンクと同じデータリンクを使用する計画である。
- 排他的 IP ゾーンによって提供されるその他の機能は必要でない。

共有 IP ゾーンには、`zonecfg` コマンドの `net` リソースを使用して、1 つ以上の IP アドレスを割
り当てます。大域ゾーンでデータリンク名も構成する必要があります。

`zonecfg net` リソースで、`address` と `physical` のプロパティを設定する必要がありま
す。`defrouter` プロパティはオプションです。

大域ゾーン内で共有 IP タイプのネットワーク接続構成を使用するには、自動ネットワーク構成ではなく `ipadm` を使用する必要があります。ネットワーク接続構成を `ipadm` によって行うかどうかを決定するには、次のコマンドを実行します。表示される応答は、`DefaultFixed` でなければなりません。

```
# svcprop -p netcfg/active_ncp svc:/network/physical:default
DefaultFixed
```

共有 IP ゾーンに割り当てられる IP アドレスは、論理ネットワークインタフェースに関連付けられます。

大域ゾーンから `ipadm` コマンドを使用すると、稼働中のゾーンの論理インタフェースの割り当てまたは削除を行うことができます。

インタフェースを追加するには、次のコマンドを使用します。

```
global# ipadm set-addrprop -p zone=my-zone net0/addr1
```

インタフェースを削除するには、次のコマンドのいずれかを使用します。

```
global# ipadm set-addrprop -p zone=global net0/addr
```

または

```
global# ipadm reset-addrprop -p zone net0/addr1
```

詳細は、『[Oracle Solaris ゾーンの実装と使用](#)』の「[共有 IP ネットワークインタフェース](#)」を参照してください。

排他的 IP 非大域ゾーン

排他的 IP は、非大域ゾーンのデフォルトのネットワーク接続構成です。

排他的 IP ゾーンには、独自の IP に関連付けられた状態と 1 つ以上の専用のデータリンクが保持されます。

排他的 IP ゾーン内では、次の各機能を使用できます。

- DHCPv4 および IPv6 ステートレスアドレスの自動構成
- IP フィルタ。ネットワークアドレス変換 (NAT) 機能も含む
- IP ネットワークマルチパス (IPMP)
- IP ルーティング
- TCP/UDP/SCTP および IP/ARP レベルのチューニング可能パラメータを設定するための `ipadm`

- IP セキュリティー (IPsec) と Internet Key Exchange (IKE)。これは、IPsec セキュリティーアソシエーション用の認証済み鍵材料のプロビジョニングを自動化する

排他的 IP ゾーンを構成する方法には、次の 2 つがあります。

- `zonecfg` ユーティリティーの `anet` リソースを使用して、ゾーンのブート時にゾーンに対して一時的な VNIC を自動的に作成し、またゾーンの停止時にそれを削除します。
- 大域ゾーン内にデータリンクを事前構成し、排他的 IP ゾーンにそれを割り当てます。これは、`zonecfg` ユーティリティーの `net` リソースを使用して行います。データリンクは、`net` リソースの `physical` プロパティを使用して指定します。`physical` プロパティは VNIC でもかまいません。`net` リソースの `address` プロパティは設定されません。

排他的 IP ゾーンは、デフォルトでは関連付けられたインタフェース上のすべての IP アドレスを構成および使用できます。必要に応じて、`allowed-address` プロパティを使用して IP アドレスのコンマ区切りリストを指定することもできます。排他的 IP ゾーンは、`allowed-address` リスト内にない IP アドレスを使用することはできません。また、ゾーンのブート時に、`allowed-address` リスト内のすべてのアドレスが排他的 IP ゾーンに対して自動で持続的に構成されます。このインタフェース構成が不要な場合、`configure-allowed-address` プロパティを `false` に設定する必要があります。デフォルト値は `true` です。

データリンクを割り当てると `snoop` コマンドが使用可能になります。

`dladm` コマンドを `show-linkprop` サブコマンドとともに使用して、実行中の排他的 IP ゾーンに対するデータリンクの割り当てを表示できます。`dladm` コマンドを `set-linkprop` サブコマンドとともに使用すると、実行中のゾーンに対して追加のデータリンクを割り当てることができます。使用例については、『Oracle Solaris ゾーンの作成と使用』の「排他的 IP 非大域ゾーンでのデータリンクの管理」を参照してください。

独自のデータリンクセットが割り当てられた、実行中の排他的 IP ゾーン内では、`ipadm` コマンドを使用して IP を構成できます。このコマンドには、論理インタフェースの追加または削除を行う機能が備わっています。ゾーンの IP 構成は、大域ゾーン内の場合と同様に `sysconfig` インタフェース (`sysconfig(1M)` のマニュアルページに記載) を使用して設定できます。

排他的 IP ゾーンの IP 構成は、大域ゾーンから `zlogin` コマンドを使用することによってのみ表示できます。

```
global# zlogin zone1 ipadm show-addr
ADDROBJ          TYPE      STATE      ADDR
lo0/v4           static    ok         127.0.0.1/8
nge0/v4          dhcp      ok         10.134.62.47/24
lo0/v6           static    ok         ::1/128
```



```
nge0/_a          addrconf ok          fe80::2e0:81ff:fe5d:c630/10
```

非大域ゾーンでの Reliable Datagram Sockets のサポート

RDS (Reliable Datagram Sockets) IPC プロトコルは、排他的 IP 非大域ゾーンと共有 IP 非大域ゾーンの両方でサポートされています。RDSv3 ドライバは、SMF サービス `rds` として有効になっています。デフォルトでは、このサービスはインストール後に無効になります。このサービスは、適切な承認を付与されたゾーン管理者が特定の非大域ゾーン内で有効にすることができます。zlogin のあとで、`rds` をそれが実行される各ゾーン内で有効にできます。

例 2-1 非大域ゾーン内で `rds` サービスを有効にする方法

1. 排他的 IP ゾーンまたは共有 IP ゾーン内で RDSv3 サービスを有効にするには、zlogin して `svcadm enable` コマンドを実行します。

```
# svcadm enable rds
```

2. `rds` が有効になっていることを確認します。

```
# svcs rds
STATE          STIME          FMRI
online         22:50:53      svc:/system/rds:default
```

詳細は、`svcadm(1M)` のマニュアルページを参照してください。

共有 IP 非大域ゾーンと排他的 IP 非大域ゾーンのセキュリティの相違

共有 IP ゾーン内のアプリケーションは、スーパーユーザーも含め、`zonecfg` ユーティリティーを介してゾーンに割り当てられた IP アドレス以外をソース IP アドレスとしてパケットを送信することはできません。このタイプのゾーンには、任意のデータリンク (レイヤー 2) パケットを送受信するアクセス権はありません。

一方、排他的 IP ゾーンの場合は、`zonecfg` によって指定されたデータリンク全体がゾーンに対して許可されます。結果として、排他的 IP ゾーン内では、スーパーユーザーまたは必要な権利プロファイルを持つユーザーは、大域ゾーン内と同様に、それらのデータリンク上でなりすましのパケットを送信できます。IP アドレスのなりすましは、`allowed-address` プロパティを設定することによって使用不可にできます。`anet` リソースの場合、`link-protection` プロパティを設定することによって、`mac-nospoof` や `dhcp-nospoof` などの追加の保護を有効にできます。

共有 IP 非大域ゾーンと排他的 IP 非大域ゾーンの同時使用

共有 IP ゾーンは常に IP 層を大域ゾーンと共有し、排他的 IP ゾーンは常に独自の IP 層インスタンスを持っています。共有 IP ゾーンと排他的 IP ゾーンの両方を同じマシンで使用することができます。

ゾーンでマウントされるファイルシステム

各ゾーンには、デフォルトでそのゾーンに委任された ZFS データセットが保持されています。このデフォルトの委任されたデータセットは、デフォルトの大域ゾーンのデータセットレイアウトのデータセットレイアウトを模倣します。.../rpool/ROOT というデータセットは、ブート環境を含んでいます。このデータセットは、直接操作しないでください。存在する必要がある rpool データセットは、デフォルトで .../rpool にマウントされます。.../rpool/export、および .../rpool/export/home データセットは /export と /export/home にマウントされます。これらの非大域ゾーンのデータセットは、対応する大域ゾーンのデータセットと使用方法が同じで、同様に管理できます。ゾーン管理者は、.../rpool、.../rpool/export、および .../rpool/export/home のデータセット内に追加のデータセットを作成できます。

[zfs\(1M\)](#) のマニュアルページで説明されている `zfs` コマンドを使用して、ゾーンの `rpool/ROOT` ファイルシステムから始まる階層内でファイルシステムの作成、削除、または名前変更を行うべきではありません。zfs コマンドは、`canmount`、`mountpoint`、`sharesmb`、`zoned`、`com.oracle.*:*`、`com.sun.*`、および `org.opensolaris.*.*` 以外のプロパティを設定するために使用できます。

通常、ゾーンでマウントされるファイルシステムには、次のものが含まれます。

- 仮想プラットフォームの初期化時にマウントされる一連のファイルシステム
- アプリケーション環境自体の内部からマウントされる一連のファイルシステム

これらのセットには、たとえば次のようなファイルシステムが含まれます。

- `none` または `legacy` 以外の `mountpoint` があり、また `canmount` プロパティに `yes` の値が保持された ZFS ファイルシステム。
- ゾーンの `/etc/vfstab` ファイルで指定されたファイルシステム。
- AutoFS によるマウントおよび AutoFS によってトリガーされるマウント。`autofs` プロパティは、`sharectl` ([sharectl\(1M\)](#) に記載) を使用して設定します。

■ ゾーン管理者が明示的に実行するマウント

また、稼働中のゾーン内でのファイルシステムのマウント権限は、`zonecfg fs-allowed` プロパティで定義します。このプロパティは、`zonecfg add fs` リソースまたは `add dataset` リソースを使用してゾーンにマウントされたファイルシステムには適用されません。デフォルトでは、ゾーンのデフォルトの委任されたデータセットである `hsfs` ファイルシステム内でのファイルシステム、および NFS などのネットワークファイルシステムのマウントだけが、ゾーン内で許可されます。



注意 - アプリケーション環境内部から実行される、デフォルト以外のマウントには、いくつかの制限事項があります。これらの制限事項は、ほかのゾーンに悪影響を与えないようにするために、ゾーン管理者がシステムのほかの部分に対するサービスを拒否できないようにします。

一部のファイルシステムについては、ゾーン内部からマウントする場合にセキュリティ制限があります。ほかのファイルシステムは、ゾーン内でマウントされたときに特有の動作を行います。詳細は、『Oracle Solaris ゾーンの作成と使用』の「ファイルシステムと非大域ゾーン」を参照してください。

データセットの詳細は、[datasets\(5\)](#) のマニュアルページを参照してください。BE の詳細は、『Oracle Solaris 11.2 ブート環境の作成と管理』を参照してください。

ファイルシステムのマウントと更新

[pkg\(5\)](#) のマニュアルページで説明されているように、ゾーンのシステムイメージの一部であるファイル、シンボリックリンク、またはディレクトリを非表示にしてファイルシステムをマウントする方法はサポートされていません。`/usr/local` に内容を提供するパッケージがインストールされていない場合は、`/usr/local` にファイルシステムをマウントすることが許容されます。しかし、いずれかのパッケージ (SVR4 パッケージを含む) が `/usr/local` で始まるパスにファイル、ディレクトリ、またはシンボリックリンクを提供する場合は、`/usr/local` へのファイルシステムのマウントはサポートされません。`/mnt` へのファイルシステムの一時的なマウントはサポートされています。

ゾーン内でファイルシステムがマウントされる順序のため、`/export` がゾーンの `rpool/export` データセットまたは別の委任されたデータセットに由来する場合は、`fs` リソースが `/export/filesys` にファイルシステムをマウントすることはできません。

ゾーン内のホスト ID

非大域ゾーンでは、大域ゾーンの `hostid` とは異なる `hostid` プロパティを設定できます。これはたとえば、別のシステム上のゾーンに移行されたマシンなどの場合に行います。このゾーン内に現在あるアプリケーションは、元の `hostid` に依存している場合があります。詳細は、[62 ページの「リソースタイプとプロパティ」](#)を参照してください。

非大域ゾーンの `/dev` ファイルシステム

`zonecfg` コマンドは、規則照合方式を使って、特定のゾーンにどのデバイスを配置するかを指定します。いずれかのルールに一致するデバイスは、ゾーンの `/dev` ファイルシステムに追加されます。詳細は、『[Oracle Solaris ゾーンの作成と使用](#)』の「[ゾーンの構成方法](#)」を参照してください。

非大域ゾーンのリムーバブル `lofi` デバイス

リムーバブルループバックファイル `lofi` デバイスは、CD-ROM デバイスのように機能し、非大域ゾーンに構成できます。デバイスがマップするファイルを変更したり、複数の `lofi` デバイスを作成して、同じファイルを読み取り専用モードで使用したりできます。このタイプの `lofi` デバイスを作成するには、`lofiadm` コマンドを `-r` オプションとともに使用します。作成時にファイル名は必要ありません。リムーバブル `lofi` デバイスのライフサイクル中は、ファイルを空のデバイスに関連付けたり、空でないデバイスからファイルの関連付けを解除したりできます。1 つのファイルを同時に複数のリムーバブル `lofi` デバイスに安全に関連付けることができます。リムーバブル `lofi` デバイスは読み取り専用です。通常の読み取り/書き込み `lofi` デバイスまたはリムーバブル `lofi` デバイスにマップされたファイルを再マッピングすることはできません。潜在的な `lofi` デバイスの数は、大域ゾーンで `zonecfg(1M)` を使用して設定できる `zone.max-lofi` リソース制御によって制限されます。

作成後のリムーバブル `lofi` デバイスは読み取り専用です。リムーバブル `lofi` デバイスに対して書き込み操作が行われると、`lofi` ドライバはエラーを返します。

`lofiadm` コマンドは、リムーバブル `lofi` デバイスを一覧表示する場合にも使用します。

例 2-2 関連するファイルを含むリムーバブル `lofi` デバイスを作成する

```
# lofiadm -r /path/to/file
```

```
/dev/lofi/1
```

例 2-3 空のリムーバブル lofi デバイスを作成する

```
# lofiadm -r
/dev/lofi/2
```

例 2-4 リムーバブル lofi デバイスにファイルを挿入する

```
# lofiadm -r /path/to/file /dev/lofi/1
/dev/lofi/1
```

詳細は、[lofiadm\(1M\)](#)、[zonecfg\(1M\)](#)、および [Lofi\(7D\)](#) のマニュアルページを参照してください。表2-2「ゾーン規模のリソース制御」も参照してください。

非大域ゾーン内でのディスク形式のサポート

ディスクのパーティション分割と `uscsi` コマンドの使用は、`zonecfg` ツールを使用すると有効になります。例については、Resource Type Propertiesの [68 ページ](#)の「[リソースタイプのプロパティ](#)」を参照してください。`uscsi` コマンドの詳細は、[uscsi\(7I\)](#) を参照してください。

- 委任は、`solaris` ゾーンでサポートされるだけです。
- ディスクは、`-D` オプションを指定した `prtconf` コマンドを使用して表示される `sd` ターゲットを使用する必要があります。[prtconf\(1M\)](#) を参照してください。

ストレージ URI を持つカーネルゾーンのデバイスリソース

次のサポートを利用できます。

- `solaris-kz` は、デバイスリソースで `bootpri` および `id` プロパティをサポートします。
 - ゾーンのルートプールの一部になるディスクでは、`bootpri` のみを設定します。ゾーンのルートプールの一部にならないディスクで `bootpri` を設定した場合、ディスク上のデータが破損する可能性があります。
 - `id` は、カーネルゾーンでディスクのインスタンスを制御します。たとえば、`id=5` は、ディスクがゾーン内で `c1d5` になることを意味します。
- ブータブル `solaris-kz` ディスクで作成されるルート `zpool` は、インストール中に大域ゾーンにインポートできます。この時点で、`zpool` コマンドを使用してルート `zpool` を表示できます。詳細は、[zpool\(1M\)](#) を参照してください。

構成可能な特権

ゾーンをブートすると、*safe* 特権のデフォルトセットが構成に含められます。これらの特権は、ゾーン内の特権プロセスがシステムのほかの非大域ゾーン内のプロセスや大域ゾーン内のプロセスに影響を及ぼすことを防ぐため、安全と見なされます。 `zonecfg` コマンドを使用して、次の操作を実行できます。

- デフォルトの特権セットに追加します。ただし、この種の変更を行うと、あるゾーン内のプロセスがグローバルリソースを制御できるようになって、ほかのゾーン内のプロセスに影響する場合があります。
- デフォルトの特権セットから削除します。ただし、この種の変更を行うと、実行に必要な特権がないため一部のプロセスが正しく動作しなくなる場合があります。

注記 - わずかですが、この時点でゾーンのデフォルト特権セットから削除できない特権があります。同じように、特権セットに追加できない特権もあります。

詳細は、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」、および `privileges(5)` を参照してください。

リソースプールの関連付け

『Oracle Solaris 11.2 でのリソースの管理』の第 13 章「リソースプールの作成と管理のタスク」の説明に従ってシステムでリソースプールを構成した場合、ゾーンの構成時に `pool` プロパティを使用して、リソースプールの 1 つにゾーンを関連付けることができます。

`dedicated-cpu` リソースを使用すると、非大域ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用割り当てよう指定できます。`dedicated-cpu` プロパティを使用すると、CPU、コア、およびソケットをゾーンに割り当てることができます。ゾーンの実行中に使用される一時プールが動的に作成されます。`zonecfg` によって指定すると、移行時にプールの設定が伝達されます。Oracle Solaris カーネルゾーンを構成する場合、`virtual-cpu` リソースも参照してください。

`pool` プロパティを使用すると、同じプールを共有する複数のゾーンを構成できます。

注記 - `pool` プロパティによって設定される持続的プールを使用するゾーン構成と、`dedicated-cpu` リソースによって構成される一時プールには、互換性はありません。これら 2 つのプロパティは、どちらか 1 つしか設定できません。

ゾーン規模のリソース制御の設定

大域管理者または適切な承認を持つユーザーは、ゾーン規模の特権付きリソース制御をゾーンに対して設定できます。ゾーン規模のリソース制御は、ゾーン内のすべてのプロセスエンティティによる総リソース消費を制限します。

これらの制限は、大域ゾーンと非大域ゾーンのどちらに対しても、`zonectfg` コマンドを使用して指定します。『Oracle Solaris ゾーンの実装と使用』の「ゾーンの構成方法」を参照してください。

ゾーン規模のリソース制御を設定するためのより簡単な推奨される方法は、`cpu-cap` などの `rctl` リソースではなく、`capped-cpu` などのプロパティ名またはリソースを使用することです。

`zone.cpu-cap` リソース制御は、1 つのゾーンで消費可能な CPU リソースの量に対する絶対的な制限を設定します。設定と同様、`100` の値は 1 つの CPU の 100% を意味します。`125` の値は 125% になります。CPU キャップの使用時は、100% がシステム上の 1 つの CPU の上限となります。

注記 - `capped-cpu` リソースを設定する場合は、単位に小数を使用できます。この値は `zone.cpu-cap` リソース制御と相互に関連していますが、設定値はその 100 分の 1 になります。設定値 `1` はリソース制御の設定値 `100` に等しくなります。

`zone.cpu-shares` リソース制御は、公平配分スケジューラ (FSS) の CPU 配分の制限をゾーンに対して設定します。CPU 配分は、まずゾーンに対して割り当てられたあとで、`project.cpu-shares` エントリの指定に従って、ゾーン内のプロジェクトに分配されます。詳細は、『Oracle Solaris ゾーンの実装と使用』の「ゾーンがインストールされている Oracle Solaris システムでの公平配分スケジューラの使用」を参照してください。この制御のグローバルプロパティ名は `cpu-shares` です。

`zone.max-locked-memory` リソース制御は、1 つのゾーンで使用できる、ロックされた物理メモリーの量に制限を設定します。ゾーン内のプロジェクト間でのロックされたメモリーリソースの割り当ては、`project.max-locked-memory` リソース制御を使用して制御できます。詳細は、『Oracle Solaris 11.2 でのリソースの管理』の「使用可能なリソース制御」を参照してください。

`zone.max-lofi` リソース制御は、1 つのゾーンで作成可能な `lofi` デバイスの数を制限します。

1 つのゾーンの LWP が多くなりすぎると、ほかのゾーンに影響を与えることがあります。`zone.max-lwps` リソース制御は、これを防ぐことで、リソースの隔離を向上させます。ゾーン内

のプロジェクト間での LWP リソースの割り当ては、`project.max-lwps` リソース制御を使用して制御できます。詳細は、『Oracle Solaris 11.2 でのリソースの管理』の「使用可能なリソース制御」を参照してください。この制御のグローバルプロパティ名は `max-lwps` です。

`zone.max-processes` リソース制御は、1 つのゾーン内のあまりに多くの LWP が他のゾーンに影響を与えることを防ぐことによって、リソースの隔離性を高めます。ゾーン内の各プロジェクトに対するプロセステーブルスロットリソースの割り当ては、`project.max-processes` リソース制御（『Oracle Solaris 11.2 でのリソースの管理』の「使用可能なリソース制御」に記載）を使用して設定できます。この制御のグローバルプロパティ名は `max-processes` です。`zone.max-processes` リソース制御は、`zone.max-lwps` リソース制御を含むこともできます。`zone.max-processes` が設定されていて `zone.max-lwps` が設定されていない場合、`zone.max-lwps` はゾーンのブート時に暗黙的に `zone.max-processes` の値の 10 倍に設定されます。正常なプロセスとゾンビプロセスの両方がプロセステーブルスロットを占有するため、`max-processes` 制御がプロセステーブルを消費しつくすゾンビプロセスから保護します。ゾンビプロセスには本来 LWP がないため、`max-lwps` でこの可能性から保護することはできません。

`zone.max-msg-ids`、`zone.max-sem-ids`、`zone.max-shm-ids`、および `zone.max-shm-memory` の各リソース制御は、ゾーン内のすべてのプロセスで使用される System V リソースを制限するために使用されます。ゾーン内のプロジェクト間での System V リソースの割り当ては、これらのリソース制御の `project` バージョンを使用して制御できます。これらの制御のグローバルプロパティ名は、`max-msg-ids`、`max-sem-ids`、`max-shm-ids`、および `max-shm-memory` です。

`zone.max-swap` リソース制御は、ゾーン内のユーザープロセスのアドレス空間マッピングと `tmpfs` マウントで消費されるスワップを制限します。`prstat -z` の出力は「スワップ」列を表示します。報告されるスワップは、ゾーンのプロセスと `tmpfs` マウントで消費されるスワップの合計量です。この値により、各ゾーンで予約されているスワップをモニタリングしやすくなり、適切な `zone.max-swap` 設定を選択できます。

表 2-2 ゾーン規模のリソース制御

| 制御名 | グローバルプロパティ名 | 説明 | デフォルトの単位 | 使用される値 |
|---------------------------|-------------|-----------------------------|---|--------|
| <code>zone.cpu-cap</code> | | このゾーンに対する CPU リソース量の絶対的な制限。 | 数量 (CPU の数)、パーセントで表されます。 注記 - <code>capped-cpu</code> リソースとして設定する場合は、単位に | |

| 制御名 | グローバルプロパティ名 | 説明 | デフォルトの単位 | 使用される値 |
|------------------------|----------------|---|---------------------|------------------------------|
| | | | 小数を使用できます。 | |
| zone.cpu-shares | cpu-shares | このゾーンに対する公平配分スケジューラ (FSS) の CPU 配分 | 数量 (配分) | |
| zone.max-locked-memory | | ゾーンで使用できるロックされた物理メモリの合計量。 priv_proc_lock_memory がゾーンに割り当てられている場合、そのゾーンがすべてのメモリをロックするのを防ぐため、このリソース制御の設定も検討してください。 | サイズ (バイト) | capped-memory の locked プロパティ |
| zone.max-lofi | max-lofi | 1 つのゾーンで作成可能な lofi デバイスの数に対する制限 | 数量 (lofi デバイスの数) | |
| zone.max-lwps | max-lwps | このゾーンで同時に使用できる LWP の最大数 | 数量 (LWP 数) | |
| zone.max-msg-ids | max-msg-ids | このゾーンに許容されるメッセージキュー ID の最大数 | 数量 (メッセージキュー ID の数) | |
| zone.max-processes | max-processes | このゾーンで同時に使用できるプロセステーブルスロットの最大数 | 数量 (プロセステーブルスロット数) | |
| zone.max-sem-ids | max-sem-ids | このゾーンに許容されるセマフォ ID の最大数 | 数量 (セマフォ ID の数) | |
| zone.max-shm-ids | max-shm-ids | このゾーンに許容される共有メモリ ID の最大数 | 数量 (共有メモリ ID の数) | |
| zone.max-shm-memory | max-shm-memory | このゾーンに許容される System V 共有メモリの合計量 | サイズ (バイト) | |
| zone.max-swap | | このゾーンのユーザープロセスのアドレス空間マッピングと tmpfs マウ | サイズ (バイト) | capped-memory の swap プロパティ |

| 制御名 | グローバルプロパティ名 | 説明 | デフォルトの単位 | 使用される値 |
|-----|-------------|-------------------|----------|--------|
| | | ントで消費できるスワップの合計量。 | | |

prctl コマンドを使用すると、実行中のプロセスに対してこれらの制限を指定できます。この例は『Oracle Solaris ゾーンの作成と使用』の「prctl コマンドを使用して大域ゾーンの FSS 配分を設定する方法」に記載されています。prctl コマンドで指定された制限には持続性はありません。システムがリブートされると、制限は無効になります。

ゾーンのコメントの追加

attr リソースの型を使ってゾーンのコメントを追加できます。詳細は、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」を参照してください。

zonecfg コマンドの使用

zonecfg コマンド (zonecfg(1M) のマニュアルページに記載) は、非大域ゾーンを構成するために使用します。

zonecfg コマンドは、大域ゾーンのリソース管理設定を持続的に指定するためにも使用できます。たとえば、このコマンドを使用して、dedicated-cpu リソースを使って専用の CPU を使用するように大域ゾーンを構成できます。

zonecfg コマンドは、対話型モード、コマンド行モード、またはコマンドファイルモードで使用できます。このコマンドを使用して、次の操作を実行できます。

- ゾーン構成を作成または削除 (破棄) します
- 特定の構成にリソースを追加します
- 構成に追加したリソースのプロパティを設定します
- 特定の構成からリソースを削除します
- 構成の照会または確認を行います
- 構成を確定します
- 前の構成に戻します
- ゾーンの名前を変更します

- zonecfg のセッションを終了します

zonecfg のプロンプトは次のような形式です。

```
zonecfg:zonename>
```

ファイルシステムなど、特定のリソースタイプの構成を行うときは、そのリソースタイプもプロンプトに表示されます。

```
zonecfg:zonename:fs>
```

この章で説明されているさまざまな zonecfg コンポーネントの使用法を示す手順を含む詳細は、『Oracle Solaris ゾーンの実装と使用』の第 1 章「非大域ゾーンの計画および構成方法」を参照してください。

zonecfg のモード

このユーザーインターフェイスでは、有効範囲という概念が使用されます。有効範囲は、大域またはリソース固有のいずれかです。デフォルトの有効範囲は大域です。

大域有効範囲で add サブコマンドまたは select サブコマンドを使用すると、特定のリソースが選択されます。すると、有効範囲がそのリソースタイプに変わります。

- add サブコマンドの場合、end、cancel のいずれかのサブコマンドを使用すると、リソースの指定が完了します。
- select サブコマンドの場合、end、cancel のいずれかのサブコマンドを使用すると、リソースの変更が完了します。

すると、有効範囲が大域に戻ります。

add、remove、set などのように、有効範囲によって異なる意味を持つサブコマンドもあります。

zonecfg の対話型モード

対話型モードでは、次のサブコマンドがサポートされます。サブコマンドの意味と使用するオプションの詳細は、zonecfg(1M) のマニュアルページを参照してください。破壊的な操作や作業内容の消失を伴うようなサブコマンドの場合、処理を実行する前にユーザーの確認が求められます。-F (強制) オプションを使用すると、この確認手順を省略できます。

| | |
|--------|---|
| help | <p>一般ヘルプまたは特定のリソースに関するヘルプを表示します。</p> <pre>zonecfg:my-zone:capped-cpu> help</pre> |
| create | <p>指定された新しいゾーンに使用するメモリー内構成の構成を開始します。次のような目的に使用されます。</p> <ul style="list-style-type: none">■ Oracle Solaris のデフォルト設定を新しい構成に適用します。この方法がデフォルトです。■ <code>-t template</code> オプションを使用して、指定したテンプレートと同一の構成を作成します。ゾーン名がテンプレート名から新しいゾーン名に変更されます。■ <code>-F</code> オプションを使用して、既存の構成を上書きします。■ <code>-b</code> オプションを使用して、なにも設定されていない空の構成を作成します。 |
| export | <p>標準出力または指定された出力ファイルに、コマンドファイルに使用できる形式で構成を出力します。</p> |
| add | <p>大域有効範囲では、指定されたリソースタイプを構成に追加します。リソース固有の有効範囲では、指定された名前と値を持つプロパティを追加します。</p> <p>詳細は、『Oracle Solaris ゾーン の作成と使用』の「ゾーンの構成方法」および <code>zonecfg(1M)</code> のマニュアルページを参照してください。</p> |
| set | <p>指定されたプロパティ名を、指定されたプロパティ値に設定します。<code>zonepath</code> などの大域的なプロパティと、リソース固有のプロパティがあることに注意してください。このコマンドは、大域有効範囲とリソース固有の有効範囲の両方で使用できます。</p> |
| select | <p>大域有効範囲でのみ使用できます。指定されたタイプのリソースのうち、指定されたプロパティ名とプロパティ値の対の条件に一致するものを、変更対象として選択します。有効範囲がそのリソースタイプに変わります。リソースが一意に識別されるように、プロパティの名前と値の対を十分な数だけ指定する必要があります。</p> |
| clear | <p>省略可能な設定の値をクリアします。必須の設定はクリアできません。ただし、必須の設定のいくつかは、新しい値を割り当てることによって変更できます。プロパティで <code>clear</code> コマンドを使用すると、値がプロパティのデフォルト値にクリアされます。</p> |
| remove | <p>大域有効範囲では、指定されたリソースタイプを削除します。リソースタイプが一意に識別されるように、プロパティの名前と値の対を十分な数だけ指定する必要があります。プロパティの名前と値の対をまったく指</p> |

| | |
|--------|--|
| | <p>定しないと、すべてのインスタンスが削除されます。該当するものが複数ある場合は、-F オプションを使用していない限り、確認を求めるメッセージが表示されます。</p> <p>リソース固有の有効範囲では、指定された名前と値を持つプロパティを現在のリソースから削除します。</p> |
| end | <p>リソース固有の有効範囲でのみ使用できます。リソースの指定を終了します。</p> <p>次に、zonecfg コマンドは、現在のリソースが正しく指定されているかどうかを確認します。</p> <ul style="list-style-type: none"> ■ リソースが正しく指定されている場合は、そのリソースがメモリー内に保持される構成に追加され、有効範囲が大域に戻ります。 ■ 指定が不完全な場合は、必要な作業を示すエラーメッセージが表示されます。 |
| cancel | <p>リソース固有の有効範囲でのみ使用できます。リソースの指定を終了し、有効範囲を大域に戻します。リソースの指定が不完全な場合、そのリソースは保持されません。</p> |
| delete | <p>指定された構成を破棄します。メモリーと安定した記憶領域の両方から構成を削除します。delete に -F (強制) オプションを使用する必要があります。</p> |



注意 - この操作は即時に実行されます。確定手順は行われず、削除されたゾーンを元に戻すことはできません。

| | |
|--------|---|
| info | <p>現在の構成または大域のリソースプロパティ zonepath、autoboot、および pool に関する情報を表示します。リソースタイプが指定されている場合は、そのタイプのリソースについてのみ情報を表示します。リソース固有の有効範囲では、このサブコマンドは、追加または変更しようとしているリソースにのみ適用されます。</p> |
| verify | <p>現在の構成が正しいかどうかを確認します。各リソースに必須プロパティがすべて指定されていることを確認します。rootzpool リソースグループとそのプロパティの構文を確認します。URI によって指定されているストレージのアクセス可能性は確認されません。</p> |
| commit | <p>現在の構成をメモリーから安定した記憶領域に確定します。メモリー内の構成を確定するまでは、revert サブコマンドで変更内容を削除できません。zoneadm で構成を使用するには、その構成を確定する必要があります。zonecfg セッションを完了するときに、この操作の実行が自動的に試みられます。正しい構成のみ確定できるので、確定操作では自動的に確認も行われます。</p> |

| | |
|--------|---|
| revert | 構成を最後に確定されたときの状態に戻します。 |
| exit | <p>zoncfg のセッションを終了します。exit に -F (強制) オプションを使用できます。</p> <p>必要な場合は、commit 操作が自動的に試行されます。EOF 文字を使ってセッションを終了することもできることに注意してください。</p> |

zoncfg のコマンドファイルモード

コマンドファイルモードでは、ファイルから入力されます。このファイルを生成するには、zoncfg Interactive Modeで説明されている [59 ページの「zoncfg の対話型モード」](#) サブコマンドを使用します。構成を標準出力に出力するか、-f オプションで指定した出力ファイルに出力することができます。

ゾーン構成データ

ゾーン構成データは、「リソース」と「プロパティ」という 2 種類のエンティティから成ります。各リソースは、タイプのほかにも 1 つ以上のプロパティを持つことがあります。プロパティは名前と値から成ります。どのようなプロパティセットを持つかは、リソースタイプによって異なります。

必須プロパティは、zonename と zonepath だけです。

リソースタイプとプロパティ

リソースとプロパティのタイプの説明は、次のとおりです。

| | |
|----------|--|
| zonename | <p>ゾーンの名前。ゾーン名には次のような規則が適用されます。</p> <ul style="list-style-type: none"> ■ 各ゾーンの名前は一意でなければならない。 ■ ゾーン名では大文字と小文字が区別される。 ■ ゾーン名は英数字で始まる必要がある。 <p>名前には、英数字、下線 (<u> </u>)、ハイフン (-)、およびピリオド (.) を使用できます。</p> <ul style="list-style-type: none"> ■ 名前の長さは 63 文字以内でなければならない。 ■ 名前 global は、大域ゾーン用に予約されています。 |
|----------|--|

- SYS で始まる名前は予約されており、使用できません。

zonepath

zonepath プロパティは、ゾーンがインストールされるパスを指定します。各ゾーンには、ルートディレクトリのパスが設定されます。これは、大域ゾーンのルートディレクトリに対する相対パスです。インストール時には、大域ゾーンのディレクトリの可視性が制限されている必要があります。ゾーンのパスの所有者は root で、モードは 700 であることが必要です。ゾーンのパスが存在しない場合、インストール時に自動的に作成されません。アクセス権が正しくない場合、自動的に修正されます。

非大域ゾーンのルートパスは 1 つ下のレベルになります。ゾーンのルートディレクトリの所有権とアクセス権は、大域ゾーンのルートディレクトリ (/) と同じになります。ゾーンのディレクトリの所有者は root で、モードは 755 であることが必要です。この階層構造により、大域ゾーンのユーザーでも権限を持っていない場合は、非大域ゾーンのファイルシステムと行き来できなくなります。

ゾーンは、ZFS データセット上に存在する必要があります。ZFS データセットは、ゾーンのインストール時または接続時に自動的に作成されません。ZFS データセットを作成できない場合、ゾーンのインストールや接続は行われません。

| パス | 説明 |
|---------------------|------------------|
| /zones/my-zone | zonecfg zonepath |
| /zones/my-zone/root | ゾーンのルート |

詳細は、『Oracle Solaris ゾーンの実装と使用』の「ファイルシステムの行き来」を参照してください。

zonecfg template プロパティでは、zonepath のデフォルト値は /system/zones/zonename です。

注記 - zoneadm の move サブコマンドで新しいフルパス zonepath を指定することにより、ゾーンを同じシステム上の別の場所に移動できます。手順については、『Oracle Solaris ゾーンの実装と使用』の「非大域ゾーンの移動」を参照してください。

autoboot

このプロパティを true に設定すると、大域ゾーンのブート時にこのゾーンが自動的にブートされます。これは、デフォルトでは false に設定されます。ゾーンサービス svc:/system/zones:default が無効になっている場合、このプロパティの設定に関係なく、ゾーンは自動的にブートしません。svcadm(1M) のマニュアルページに記載されているように、svcadm コマンドを使用してゾーンサービスを有効にできます。

```
global# svcadm enable zones
```

pkg update を実行中のこの設定については、『Oracle Solaris ゾーンの作成と使用』の「ゾーンのパッケージングの概要」を参照してください。

| | |
|------------------|---|
| bootargs | <p>このプロパティは、ゾーンのブート引数を設定するために使用しません。reboot、zoneadm boot、または zoneadm reboot コマンドで無効にされた場合を除き、このブート引数が適用されます。Zone Boot Argumentsを参照してください。</p> |
| limitpriv | <p>このプロパティは、デフォルト以外の特権マスクを指定するために使用します。『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」を参照してください。</p> <p>特権を追加するには、特権名だけを指定するか、特権名の前に priv_ 付けて指定します。特権を除外するには、名前の前にダッシュ (-) または感嘆符 (!) を付けます。複数の特権は、コンマで区切り、引用符 (") で囲みます。</p> <p>priv_str_to_set(3C) で説明されているように、特殊な特権セット none、all、および basic は、それぞれの通常の定義に展開されます。ゾーン構成は大域ゾーンで行われるため、特殊な特権セット zone は使用できません。特定の特権を追加または削除してデフォルトの特権セットを変更するのが一般的な使用方法であるため、特殊なセットである default はデフォルトの特権セットにマップされます。limitpriv プロパティの先頭に default がある場合、デフォルトセットに展開されます。</p> <p>次のエントリは、dtrace_proc 特権と dtrace_user 特権だけを必要とする DTrace プログラムをゾーンで使用できるようにします。</p> <pre>global# zonecfg -z userzone zonecfg:userzone> set limitpriv="default,dtrace_proc,dtrace_user"</pre> <p>ゾーンの特権セットに不許可の特権が含まれる場合、必須の特権が欠落している場合、または未知の特権が含まれる場合、ゾーンの検証、準備、またはブートの試行は失敗し、エラーメッセージが表示されます。</p> |
| scheduling-class | <p>このプロパティは、ゾーンのスケジューリングクラスを設定します。詳細とヒントについては、39 ページの「スケジューリングクラス」を参照してください。</p> |
| ip-type | <p>このプロパティは、すべての非大域ゾーンのために設定する必要があります。47 ページの「排他的 IP 非大域ゾーン」、46 ページの「共有 IP 非大域ゾーン」、および『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」を参照してください。</p> |
| dedicated-cpu | <p>このリソースは、ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用割り当てます。dedicated-cpu リソースは、ncpus の制限を設定し、必要に応じて importance、ncores、cores、および sockets も設</p> |

定します。詳細は、[37 ページの「dedicated-cpu リソース」](#)を参照してください。

| | |
|--------------------------------------|---|
| solaris-kz のみ: virtual-cpu | この solaris-kz リソースは、ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用に割り当てます。virtual-cpu リソースは、ncpus の制限を設定します。詳細は、 38 ページの「solaris-kz のみ: virtual-cpu リソース」 を参照してください。 |
| capped-cpu | このリソースは、ゾーンの実行中にゾーンで消費可能な CPU リソースの量に対する制限を設定します。この capped-cpu リソースは、ncpus に制限を設けます。詳細は、 38 ページの「capped-cpu リソース」 を参照してください。 |
| capped-memory | このリソースは、ゾーンのメモリー上限を設定する際に使用される各プロパティをグループ化します。capped-memory リソースは、physical、swap、および locked メモリーの制限を設定します。これらのプロパティの少なくとも 1 つは指定する必要があります。capped-memory リソースを使用するには、service/resource-cap パッケージが大域ゾーン内にインストールされている必要があります。 |
| anet | anet リソースは、排他的 IP ゾーンのブート時にゾーンに対して一時的な VNIC インタフェースを自動的に作成し、またゾーンの停止時にそれを削除します。 |
| net | net リソースは、非大域ゾーンに大域ゾーン内にある既存のネットワークインタフェースを割り当てます。ネットワークインタフェースリソースは、インタフェースの名前です。各ゾーンでは、インストール済み状態から準備完了状態に移行するときに設定される、複数のネットワークインタフェースを保持できます。 |
| dataset | <p>データセットは、ファイルシステム、ボリューム、またはスナップショットの総称です。ZFS™ データセットリソースを追加すると、ストレージ管理を非大域ゾーンに委任できるようになります。委任されたデータセットがファイルシステムの場合は、ゾーン管理者はそのデータセット内のファイルシステムの作成と破棄、およびデータセットのプロパティの変更を行うことができます。ゾーン管理者は、スナップショット、子のファイルシステムとボリューム、その子孫のクローンを作成できます。委任されたデータセットがボリュームの場合は、ゾーン管理者はプロパティの設定とスナップショットの作成を行うことができます。ゾーン管理者は、ゾーンに追加されていないデータセットを操作したり、ゾーンに割り当てられているデータセットに設定されている最上位レベルの割り当て制限を超過したりすることはできません。データセットが非大域ゾーンに委任されると、zoned プロパティが自動的に設定されます。ゾーン管理者がマウントポイントを受け入れられない値に設定しなければならない可能性があるため、ゾーンファイルシステムを大域ゾーンにマウントすることはできません。</p> <p>次の方法で、ZFS データセットをゾーンに追加できます。</p> |

- `lofs` マウントされたファイルシステムとして (大域ゾーンとの領域共有のみが目的の場合)
- 委任されたデータセットとして

`zonecfg template` プロパティの使用中に `rootzpool` リソースが指定されていない場合、デフォルトの `zonename` データセットは `rootzpool/VARSHARE/zones/zonename` です。データセットは、マウントポイント / `system/zones` を持つ `svc-zones` サービスによって作成されます。残りのプロパティは、`rootzpool/VARSHARE/zones/` から継承されます。

『Oracle Solaris 11.2 での ZFS ファイルシステムの管理』の第 9 章「Oracle Solaris ZFS の高度なトピック」、『Oracle Solaris ゾーン の作成と使用』の「ファイルシステムと非大域ゾーン」、および [datasets\(5\)](#) のマニュアルページを参照してください。

データセットの問題については、『Oracle Solaris ゾーン の作成と使用』の第 13 章「Oracle Solaris ゾーンで発生するさまざまな問題のトラブルシューティング」を参照してください。

`fs` 各ゾーンでは、インストール済み状態から準備完了状態に移行するときにマウントする各種のファイルシステムを指定できます。ファイルシステムリソースは、ファイルシステムのマウントポイントのパスを指定します。ゾーンでのファイルシステムの使用の詳細は、『Oracle Solaris ゾーン の作成と使用』の「ファイルシステムと非大域ゾーン」を参照してください。

注記 - 非大域ゾーン内の UFS ファイルシステムを `fs` リソースによって使用するには、インストールの後か、または AI マニフェストスクリプトを使用して、`system/file-system/ufs` パッケージをゾーンにインストールする必要があります。

`fs` リソースによって追加された UFS ファイルシステムの割り当て制限情報を取得するために、`quota` コマンド ([quota\(1M\)](#)) を参照) は使用できません。

`fs-allowed` このプロパティを設定すると、ゾーン管理者によって作成されたか、または NFS を使用してインポートされた、該当のタイプの任意のファイルシステムをマウントし、またそのファイルシステムを管理するための機能がゾーン管理者に提供されます。また、稼働中のゾーン内でのファイルシステムのマウント権限は、`fs-allowed` プロパティで制限します。デフォルトでは、`hsfs` ファイルシステムおよび NFS などのネットワークファイルシステムのマウントだけがゾーン内で許可されます。

このプロパティは、ゾーンに委任されたブロックデバイスまたは ZVOL デバイスでも使用できます。

`fs-allowed` プロパティは、ゾーン内からマウントできる、追加のファイルシステムのコンマ区切りリスト (たとえば、`ufs,pcfs`) を受け入れます。

```
zonecfg:my-zone> set fs-allowed=ufs,pcfs
```

このプロパティは、`add fs` プロパティまたは `add dataset` プロパティを使用して大域ゾーンによって管理されるゾーンのマウントに影響を与えません。

セキュリティに関する考慮事項については、『Oracle Solaris ゾーンの作成と使用』の「ファイルシステムと非大域ゾーン」および『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンでのデバイスの使用」を参照してください。

device `zonefigdevice` リソースは、非大域ゾーンのプラットフォームに仮想ディスクを追加するために使用します。デバイスリソースは、デバイス照合の指定子です。各ゾーンでは、インストール済み状態から準備完了状態に移行するときに構成するべきデバイスを指定できます。

注記 - 非大域ゾーン内の UFS ファイルシステムを `device` リソースによって使用するには、インストールのあとか、または AI マニフェストスクリプトを使用して、`system/file-system/ufs` パッケージをゾーンにインストールする必要があります。

pool このプロパティは、システム上のリソースプールにゾーンを関連付けるために使用します。1 つのプール内のリソースを複数のゾーンが共有してもかまいません。37 ページの「`dedicated-cpu` リソース」も参照してください。

rctl `rctl` リソースは、ゾーン規模のリソース制御に使用されます。リソース制御は、ゾーンがインストール済み状態から準備完了状態に移行するときに有効になります。

詳細は、55 ページの「ゾーン規模のリソース制御の設定」を参照してください。

注記 - `rctl` リソースの代わりに `zonefig` の `set global_property_name` サブコマンドを使用してゾーン規模の制御を構成するには、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」を参照してください。

attr この汎用属性は、ユーザーコメントとして使用したり、ほかのサブシステムで使用したりできます。`attr` の `name` プロパティは、英数字で始まる必要があります。`name` プロパティには、英数字、ハイフン (-)、およびピリオド (.) を使用できます。`zone.` で始まる属性名はシステム用に予約されています。

リソースタイプのプロパティ

リソースには、構成可能なプロパティもあります。リソースタイプとそれに関連付けられるプロパティは次のとおりです。

admin ユーザー名と特定のゾーンに対するそのユーザーの承認を定義します。

```
zonecfg:my-zone> add admin
zonecfg:my-zone:admin> set user=zadmin
zonecfg:my-zone:admin> set auths=login,manage
zonecfg:my-zone:admin> end
```

auths プロパティには、次の値を使用できます。

- login (solaris.zone.login)
- manage (solaris.zone.manage)
- clone (solaris.zone.clonefrom)

これらの **auths** はゾーンの作成を可能にするわけではありません。この機能は、ゾーンのセキュリティプロファイル内に含まれています。

**solaris および
solaris10 のみ:
rootzpool**

storage

ゾーンのインストール専用の ZFS **zpool** を提供するためのストレージオブジェクトの URI を識別します。URI と **storage** に指定可能な値については、[41 ページの「solaris および solaris10 のみ: rootzpool リソース」](#)を参照してください。ゾーンのインストール時に、**zpool** が自動的に作成されるか、事前に作成された **zpool** がインポートされます。*my-zone_rpool* という名前が割り当てられます。

```
zonecfg:my-zone> add rootzpool
zonecfg:my-zone:rootzpool> add storage dev:dsk/c4t1d0
zonecfg:my-zone:rootzpool> end
```

ミラー化構成を作成する場合は、**storage** プロパティを追加できます。

```
add storage dev:dsk/c4t1d0
add storage dev:dsk/c4t3d0
```

1 つのゾーンに構成できる **rootzpool** リソースは 1 つだけです。

**solaris および
solaris10 のみ:
zpool**

storage, name

zpool をゾーンに委任するためのストレージオブジェクトの URI を 1 つ以上定義します。URI と **storage** プロパティに指定可能な値については、[41 ページの「solaris および solaris10 のみ: rootzpool リソース」](#)を参照してください。**name** プロパティに指定可能な値は、[zpool\(1M\)](#) のマニュアルページで定義されています。

この例では、zpool ストレージリソースがゾーンに委任されています。zpool は自動的に作成されるか、または以前に作成された zpool がインストール中にインポートされます。zpool の名前は my-zone_pool1 です。

```
zonecfg:my-zone> add zpool
zonecfg:my-zone:zpool> set name=pool1
zonecfg:my-zone:zpool> add storage dev:dsk/c4t2d0
zonecfg:my-zone:zpool> add storage dev:dsk/c4t4d0
zonecfg:my-zone:zpool> end
```

ゾーン構成には、1 つ以上の zpool リソースを含めることができます。

dedicated-cpu

ncpus, importance, cores, cpus, sockets

CPU の数を指定し、必要に応じてプールの相対的な重要性も指定します。次の例では、ゾーン my-zone で使用する CPU の範囲を指定します。importance も設定します。

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=1-3
zonecfg:my-zone:dedicated-cpu> set importance=2
zonecfg:my-zone:dedicated-cpu> end
```

コア 0、1、2、および 3 を永続的にゾーン my-zone に割り当てます。次の dedicated-cpu の例では cores を使用しますが、cpus=、cores=、および sockets= はすべて使用可能です。

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set cores=0-3
zonecfg:my-zone:dedicated-cpu> end
```

virtual-cpu

ncpus

CPU の数を指定します。次の例では、ゾーン my-zone に 3 つの CPU を指定します。

```
zonecfg:my-zone> add virtual-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=3
zonecfg:my-zone:dedicated-cpu> end
```

capped-cpu

ncpus

CPU の数を指定します。次の例では、ゾーン my-zone の CPU 数のキャップを 3.5 に指定します。

```
zonecfg:my-zone> add capped-cpu
zonecfg:my-zone:capped-cpu> set ncpus=3.5
zonecfg:my-zone:capped-cpu> end
```

capped-memory

physical, swap, locked

ゾーン `my-zone` のメモリー制限を指定します。各制限はオプションですが、少なくとも 1 つは設定する必要があります。

```
zonecfg:my-zone> add capped-memory
zonecfg:my-zone:capped-memory> set physical=50m
zonecfg:my-zone:capped-memory> set swap=100m
zonecfg:my-zone:capped-memory> set locked=30m
zonecfg:my-zone:capped-memory> end
```

`capped-memory` リソースを使用するには、`resource-cap` パッケージが大域ゾーン内にインストールされている必要があります。

fs

`dir`、`special`、`raw`、`type`、`options`

`fs` リソースのパラメータは、ファイルシステムをマウントする方法と場所を決定する値を指定します。`fs` のパラメータは次のように定義されています。

| | |
|----------------------|--|
| <code>dir</code> | ファイルシステムのマウントポイントを指定します |
| <code>special</code> | 大域ゾーンからマウントするブロック型特殊デバイスの名前またはディレクトリを指定します |
| <code>raw</code> | ファイルシステム (ZFS には適用されない) をマウントする前に、 <code>fsck</code> の実行対象の <code>raw</code> デバイスを指定します。 |
| <code>type</code> | ファイルシステムのタイプを指定します |
| <code>options</code> | <code>mount</code> コマンドで使用されるオプションに似たマウントオプションを指定します |

次の例の各行では、大域ゾーン内の `pool1/fs1` という名前のデータセットを、構成されるゾーン内で `/shared/fs1` としてマウントすることを指定しています。使用するファイルシステムのタイプは ZFS です。

```
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/shared/fs1
zonecfg:my-zone:fs> set special=pool1/fs1
zonecfg:my-zone:fs> set type=zfs
zonecfg:my-zone:fs> end
```

パラメータの詳細は、『Oracle Solaris ゾーンの作成と使用』の「[onosuid オプション](#)」、『Oracle Solaris ゾーンの作成と使用』の「[セキュリティの制限およびファイルシステムの動作](#)」、および [fsck\(1M\)](#) と [mount\(1M\)](#) のマニュアルページを参照してください。また、セクション 1M のマニュアルページには、特定のファイルシステムに固有のマウントオプションに関するものがあります。このようなマニュアルページの名前は、`mount_filesystem` という形式です。

注記 - このリソースを使用して追加された UFS ファイルシステムの割り当て制限情報を取得するために、`quota` コマンド ([quota\(1M\)](#) を参照) は使用できません。

| | |
|-------------|--|
| dataset | name |
| name, alias | <p>次の例の各行では、データセット <code>sales</code> を非大域ゾーンでマウントして可視にし、大域ゾーンでは不可視にするように指定しています。</p> <pre>zonecfg:my-zone> add dataset zonecfg:my-zone> set name=tank/sales zonecfg:my-zone> end</pre> <p>委任されたデータセットは、次の例に示されているようにデフォルト以外の別名を持つことができます。データセットの別名には、スラッシュ (/) を含めることができないことに注意してください。</p> <pre>zonecfg:my-zone> add dataset zonecfg:my-zone:dataset> set name=tank/sales zonecfg:my-zone:dataset> set alias=data zonecfg:my-zone:dataset> end</pre> <p>デフォルトの別名に戻すには、<code>clear alias</code> を使用します。</p> <pre>zonecfg:my-zone> clear alias</pre> |
| anet | <p>linkname, lower-link, allowed-address, auto-mac-address, configure-allowed-address, defrouter, linkmode (IPoIB), mac-address (IPoIB 以外), mac-slot (IPoIB 以外), mac-prefix (IPoIB 以外), mtu, maxbw, pkey (IPoIB), priority, vlan-id (IPoIB 以外), rxfanout, rxrings, txrings, link-protection, allowed-dhcp-cids</p> <p>solaris のみ:zonecfg では、IPoIB データリンクに次の anet プロパティを設定しないでください。</p> <ul style="list-style-type: none"> ■ mac-address ■ mac-prefix ■ mac-slot ■ vlan-id <p>zonecfg では、IPoIB 以外のデータリンクに次の anet プロパティを設定しないでください。</p> <ul style="list-style-type: none"> ■ linkmode ■ pkey <p>EVS anet リソースには、次のプロパティのみを設定してください。</p> |

- linkname
- evs
- vport
- configure-allowed-address

anet リソースは、ゾーンのブート時に自動 VNIC インタフェースまたは IPoIB インタフェースを作成し、ゾーンの停止時にその VNIC または IPoIB インタフェースを削除します。solaris-kz ブランドでは IPoIB はサポートされません。リソースプロパティは、zonecfg コマンドを使用して管理されます。使用できるプロパティのすべてのテキストについては、[zonecfg\(1M\)](#) のマニュアルページを参照してください。

lower-link 作成されるリンクの基礎となるリンクを指定します。auto に設定すると、zoneadmd デーモンは、ゾーンがブートするたびに VNIC が作成されるリンクを自動的に選択します。anet リソースの lower-link として VNIC を作成できる任意のリンクを指定できます。

ブート中に VNIC を自動的に作成するためのデータリンクを選択すると、すべての IPoIB リンクがスキップされます。

linkname 自動作成される VNIC インタフェースまたは IPoIB インタフェースの名前を指定します。solaris-kz では IPoIB はサポートされません。

mac-address (IPoIB には非適用) 指定された値またはキーワードに基づいて VNIC の MAC アドレスを設定します。値がキーワードでない場合は、ユニキャスト MAC アドレスとして解釈されます。サポートされるキーワードについては、[zonecfg\(1M\)](#) のマニュアルページを参照してください。ランダムな MAC アドレスを選択すると、生成されたアドレスはゾーンの再ブートおよびゾーンの切り離しと接続の操作後も保持されます。デフォルトのポリシー auto-mac-address を使用するときは、Oracle Solaris ゾーンはランダムな mac-address を取得できます。

pkey (IPoIB のみ) IPoIB データリンクインタフェースを作成するために使用するパーティションキーを設定します。このプロパティは必須です。指定した pkey は、0x 接

頭辞があるかどうかに関係なく、常に 16 進数として扱われます。

linkmode (IPoIB のみ)

データリンクインタフェースの linkmode を設定します。デフォルト値は cm です。有効な値は次のとおりです。

cm (デフォルト) 接続モード。このモードでは、65520 バイトのデフォルトの MTU を使用し、65535 バイトの最大 MTU をサポートします。

ud 低信頼データグラムモード。リモートノードに対して接続モードが使用できない場合は、代わりに低信頼データグラムモードが自動的に使用されます。このモードでは、2044 のデフォルトの MTU を使用し、4092 バイトの最大 MTU をサポートします。

allowed-address

排他的 IP ゾーン用の IP アドレスを構成し、また排他的 IP ゾーンで使用できる、構成可能な IP アドレスセットの制限も行います。複数のアドレスを指定するには、IP アドレスのコンマ区切りリストを使用します。

defrouter

defrouter プロパティを使用すると、非大域ゾーンと大域ゾーンが別個のネットワーク上に存在する場合にデフォルト経路を設定できます。

defrouter プロパティが設定されたゾーンは、大域ゾーン用に構成されていないサブネット上に存在する必要があります。

zonecfg コマンドで SYSdefault テンプレートを使用してゾーンを作成したときに、ほかの IP リソースが設定されていない場合は、次のプロパティを持つ anet リソースがゾーン構成に自動的に組み込まれます。linkname は、物理 Ethernet リンクに対して自動的に作成され、netN という形式の最初に使用できる名前 (net0) に設定されます。デフォルト値を変更するには、zonecfg コマンドを使用します。

デフォルトのポリシー auto を使用すると、適切な mac-address が割り当てられます。

Oracle Solaris ゾーン ランダムな mac-address

Oracle Solaris カーネルゾーン ランダムな mac-address

カーネルゾーンの下
Oracle Solaris ゾーン ファクトリ mac-address

Oracle VM Server for SPARC ゲストドメイン ファクトリ mac-address

Oracle VM Server for SPARC ゲストドメインで実行されている
Oracle Solaris カーネルゾーン ファクトリ mac-address

デフォルトのポリシーでは、物理 Ethernet リンク (たとえば、net0) に対して自動 VNIC が作成され、この VNIC に MAC アドレスが割り当てられます。オプションの lower-link プロパティは、自動 VNIC が作成される、基礎となるリンク vnic1 に設定されます。zonecfg コマンドを使用して、リンク名、ベースとなる物理リンク、MAC アドレス、帯域幅制限などの VNIC プロパティ、およびその他の VNIC プロパティを指定できます。ip-type=exclusive も指定する必要があります。

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add anet
zonecfg:my-zone:anet> set linkname=net0
zonecfg:my-zone:anet> set lower-link=auto
zonecfg:my-zone:anet> set mac-address=random
zonecfg:my-zone:anet> set link-protection=mac-nospoof
zonecfg:my-zone:anet> end
```

次の例は、物理リンク net5 上の IPoIB データリンクインタフェースを使用し、IB のパーティションキーを 0xffff にして構成された solaris ブランドゾーンを示しています。

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone:anet> add anet
zonecfg:my-zone:anet> set linkname=ib0
zonecfg:my-zone:anet> set lower-link=net5
zonecfg:my-zone:anet> set pkey=0xffff
zonecfg:my-zone:anet> end
```

プロパティの詳細は、[zonecfg\(1M\)](#) のマニュアルページを参照してください。リンクプロパティの詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

```
net                address, allowed-address, physical, defrouter
```

注記 - 共有 IP ゾーンの場合は、IP アドレスと物理デバイスの両方を指定する必要があります。必要に応じて、デフォルトのルーターを設定できます。

排他的 IP ゾーンの場合は、物理インタフェースだけを指定する必要があります。

- `allowed-address` プロパティは、排他的 IP ゾーンで使用できる、構成可能な IP アドレスのセットを制限します。
- `defrouter` プロパティを使用すると、非大域ゾーンと大域ゾーンが別個のネットワーク上に存在する場合にデフォルト経路を設定できます。
- `defrouter` プロパティが設定されたゾーンは、大域ゾーン用に構成されていないサブネット上に存在する必要があります。
- デフォルトのルーターを持つゾーンからのトラフィックは、宛先のゾーンに戻る前にそのルーターに送信されます。

共有 IP ゾーンがそれぞれ異なるサブネット上にある場合は、大域ゾーンでデータリンクを構成しないでください。

共有 IP ゾーンの場合の次の例では、物理インタフェース `nge0` が、IP アドレス `192.168.0.1` のゾーンに追加されています。システム上のネットワークインタフェースをリスト表示するには、次のように入力します。

```
global# ipadm show-if -po ifname,class,active,persistent
lo0:loopback:yes:46--
nge0:ip:yes:----
```

出力の各行には、ループバックの行を除いて、ネットワークインタフェースの名前があります。説明に `loopback` が含まれている行は、カードには当てはまりません。`46` の `persistent` フラグは、そのインタフェースが大域ゾーン内で持続的に構成されていることを示します。`yes` という有効な値は、そのインタフェースが現在構成されていることを示しています。また `class` の `ip` の値は、`nge0` がループバックインタフェースではないことを示しています。ゾーンのデフォルト経路は、`10.0.0.1` に設定されています。`defrouter` プロパティの設定はオプションです。`ip-type=shared` は必須です。

```
zonecfg:my-zone> set ip-type=shared
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=vnic1
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-zone:net> set defrouter=10.0.0.1
zonecfg:my-zone:net> end
```

排他的 IP ゾーンの場合の次の例では、VNIC が、VLAN である物理インタフェースに使用されます。使用可能なデータリンクを調べるには、`dladm show-link` を使用してください。`allowed-address` プロパティは、ゾーンが使用できる IP アドレスを抑制します。`defrouter` プロパティは、デフォルト経路を設定するために使用します。`ip-type=exclusive` も指定する必要があります。

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add net
zonecfg:myzone:net> set allowed-address=10.1.1.32/24
zonecfg:my-zone:net> set physical=vnic1
zonecfg:myzone:net> set defrouter=10.1.1.1
zonecfg:my-zone:net> end
```

`add net` 手順では、物理デバイスタイプだけを指定します。`physical` プロパティは VNIC でもかまいません。

注記 - Oracle™ Solaris オペレーティングシステムは、すべての Ethernet タイプインタフェースをサポートします。また、それらのデータリンクは `dladm` コマンドを使用して管理できます。

`device` `match, allow-partition, allow-raw-io`

マッチングするデバイス名は、マッチングするパターンまたは絶対パスにできます。`allow-partition` と `allow-raw-io` の両方を `true` または `false` に設定できます。デフォルトは `false` です。`allow-partition` はパーティション分割を有効にします。`allow-raw-io` は `uscsi` を有効にします。これらのリソースの詳細は、[zonecfg\(1M\)](#) を参照してください。

`solaris-kz` ゾーンについて `device:match` リソースプロパティで指定できる制限には、次のものが含まれます。

- LUN ごとに 1 つのリソースのみが許可されます。
- スライスとパーティションはサポートされません。
- `raw` ディスクデバイスのサポートのみが提供されます。
- サポートされるデバイスパスは、`lofi`、`ramdisk`、`dsk`、および `zvol` です。

次の例では、ディスクデバイスに対する `uscsi` 操作が `solaris` ゾーン構成に含まれています。

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set match=/dev/*dsk/cXtYdZ*
zonecfg:my-zone:device> set allow-raw-io=true
zonecfg:my-zone:device> end
```

Veritas Volume Manager デバイスは、`add device` を使用して非大域ゾーンに委任されます。

次の例では、ストレージデバイスが solaris-kz ゾーンに追加されています。

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set storage=iscsi:///
luname.naa.600144f03d70c80000004ea57da10001
zonecfg:my-zone:device> set bootpri=0
zonecfg:my-zone:device> end
```



注意 - デバイスを追加する前に、制限とセキュリティーに関する確認事項について、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンでのデバイスの使用」、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンでのアプリケーションの実行」、および『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」を参照してください。

rctl

name, value

使用可能なゾーン規模のリソース制御は、次のとおりです。

- zone.cpu-cap
- zone.cpu-shares (推奨: cpu-shares)
- zone.max-locked-memory
- zone.max-lofi
- zone.max-lwps (推奨: max-lwps)
- zone.max-msg-ids (推奨: max-msg-ids)
- zone.max-processes (推奨: max-processes)
- zone.max-sem-ids (推奨: max-sem-ids)
- zone.max-shm-ids (推奨: max-shm-ids)
- zone.max-shm-memory (推奨: max-shm-memory)
- zone.max-swap

ゾーン規模のリソース制御を設定するために推奨される、より簡単な方法は、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」に示されているように、rctl リソースの代わりにプロパティ名を使用する方法です。add rctl を使用してゾーン内のゾーン規模のリソース制御エントリを構成する場合、その形式は project データベース内のリソース制御エントリの形式とは異なります。ゾーン構成では、rctl リソースタイプは、名前と値の対 3 つから成ります。これらの名前は、priv、limit、および action です。これらの名前には、単純な値がそれぞれ設定されます。

```
zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.cpu-shares
zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
zonecfg:my-zone:rctl> end
```

```
zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.max-lwps
zonecfg:my-zone:rctl> add value (priv=privileged,limit=100,action=deny)
zonecfg:my-zone:rctl> end
```

リソース制御と属性の全般的な情報については、『Oracle Solaris 11.2 でのリソースの管理』の第 6 章「リソース制御について」および『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンで使用されるリソース制御」を参照してください。

attr **name, type, value**
次の例では、ゾーンに関するコメントを追加します。

```
zonecfg:my-zone> add attr
zonecfg:my-zone:attr> set name=comment
zonecfg:my-zone:attr> set type=string
zonecfg:my-zone:attr> set value="Production zone"
zonecfg:my-zone:attr> end
```

export サブコマンドを使用すると、ゾーン構成を標準出力に出力できます。構成は、コマンドファイルに使用できる形式で保存されます。

Tecla コマンド行編集ライブラリ

付属の Tecla コマンド行編集ライブラリは、**zonecfg** コマンドで使用できます。このライブラリにより、コマンド行の履歴メカニズムと編集サポートが提供されます。

詳細については、[tecla\(5\)](#) のマニュアルページを参照してください。

用語集

| | |
|-----------------|--|
| アカウントイン グの拡張 | Solaris オペレーティングシステムで、タスクまたはプロセスに基づくリソース消費量を柔軟に記録できる方法。 |
| 完全ルートゾ ン | 必要なすべてのシステムソフトウェアおよび追加パッケージがゾーン固有のファイルシステムにインストールされる非大域ゾーンのタイプ。 |
| 局所有効範囲 | 制御値を超えようとしているプロセスに対して行われる局所アクション。 |
| 公平配分スケ ジューラ | 公平さを基準に CPU 時間を割り当てるスケジューリングクラス (FSS と呼ぶ)。配分は、システムの CPU リソースのうちプロジェクトに割り当てる部分を定義します。 |
| 作業セットサイ ズ | 作業セットのサイズ。作業セットとは、プロジェクトの作業負荷がその処理サイクル中にアクティブに使用するページのことです。 |
| 作業負荷 | アプリケーションまたはアプリケーショングループのプロセスすべての合計。 |
| 上限制御 | システムリソース使用率を規制するプロセス。 |
| 常駐セットサイ ズ | 常駐セットのサイズ。常駐セットとは、物理メモリーに常駐するページのことです。 |
| スカナ | あまり使用されていないページを識別するカーネルスレッド。メモリーが少ない状況では、スカナは最近使用されていないページを再生します。 |
| 静的プール構 成 | 管理者が、リソースプール機能に関してシステムを構成する方法を表現したもの。 |
| ゾーン管理者 | ゾーン管理者の特権は、非大域ゾーンに対してのみ有効です。 大域管理者 も参照してください。 |
| ゾーン状態 | 非大域ゾーンのステータス。ゾーンの状態は、構成済み、不完全、インストール済み、準備完了、使用不可、稼働、または停止処理のいずれかになります。 |
| 大域管理者 | root ユーザーまたは root 役割を持つ管理者大域ゾーンにログインすると、大域管理者または適切な承認が付与されたユーザーは、システム全体をモニターしたり制御したりできます。 ゾーン管理者 も参照してください。 |

| | |
|---------------|--|
| 大域ゾーン | すべての Oracle Solaris システムに含まれるゾーン。非大域ゾーンを使用しているときには、大域ゾーンはシステムのデフォルトゾーンであると同時に、システム規模の管理制御に使用されるゾーンでもあります。 非大域ゾーン も参照してください。 |
| 大域有効範囲 | システム上のすべてのリソース制御のリソース制御値に適用されるアクション。 |
| タスク | リソース管理において、長時間にわたる作業の集合を表すプロセスの集まり。各タスクは 1 つのプロジェクトに関連付けられます。 |
| データリンク | OSI プロトコルスタックのレイヤー 2 インタフェース。システム内で STREAMS DLPI (v2) インタフェースとして表されます。このインタフェースは、TCP/IP などのプロトコルスタックで接続できます。Oracle Solaris 10 ゾーンでは、データリンクは物理インタフェース、アグリゲーション、または VLAN でタグ付けされたインタフェースです。データリンクは、物理インタフェースとして参照されることもあります (たとえば、NIC または VNIC を指す場合)。 |
| デフォルトプール | プールが有効に設定される際にシステムにより作成されるプール。 リソースプール も参照してください。 |
| デフォルトプロセッサセット | プールが有効に設定される際にシステムにより作成されるプロセッサセット。 プロセッサセット も参照してください。 |
| 動的構成 | ある時点における、指定されたシステムのリソースプールフレームワーク内部のリソース配置に関する情報。 |
| 動的再構成 | SPARC ベースのシステムで、システムの稼働中にハードウェアを再構成する機能。DR とも呼ばれます。 |
| ネームサービスデータベース | このドキュメントの「プロジェクトとタスク (概要)」の章では、LDAP コンテナと NIS マップの両方を指して使用されます。 |
| ヒープ | プロセス内で割り当てられたスクラッチメモリー。 |
| 非大域ゾーン | Oracle Solaris オペレーティングシステムの単一インスタンス内に作成された仮想オペレーティングシステム環境。Oracle Solaris ゾーンソフトウェア区分技術を使用して、オペレーティングシステムサービスが仮想化されます。 |
| 非大域ゾーン管理者 | ゾーン管理者 を参照してください。 |
| プール | リソースプール を参照してください。 |
| プールデーモン | リソースの動的割り当てが必要な場合にアクティブになる <code>poold</code> システムデーモン。 |
| ブランド | BrandZ 機能のインスタンス。アプリケーションの実行に使用される、ネイティブでないオペレーティング環境が含まれている非大域ゾーンを提供します。 |

| | |
|-------------|--|
| ブランドゾーン | ネイティブでないアプリケーションを非大域ゾーンで実行するための隔離された環境。 |
| プロジェクト | ネットワーク全体の関連作業に対する管理識別子。 |
| プロセッサセット | 互いに素である CPU のグループ。各プロセッサセットには、0 以上のプロセッサを含めることができます。プロセッサセットは、リソースプール構成内でリソース要素として表されます。これは pset とも呼ばれます。 素も参照してください。 |
| 補助ゾーン状態 | 追加の状態情報を大域ゾーンに伝えるために使用されます。ゾーン状態も参照してください。 |
| メモリ上限実行しきい値 | リソース上限デーモンが上限を制限するときのシステム上における物理メモリーの使用効率 (パーセンテージ)。 |
| 素 | セットのメンバーが重複しないセットのタイプ。 |
| 読み取り専用ゾーン | 読み取り専用ルートを持つように構成された不変ゾーン。 |
| リソース | アプリケーションの動作を変更する目的で操作されるコンピューティングシステムの側面。 |
| リソース管理 | アプリケーションが利用可能なシステムリソースをどのように使用するかを制御する機能。 |
| リソース上限デーモン | リソース上限が定義されたプロジェクト内で動作するプロセスが消費する物理メモリーを規制するデーモン。 |
| リソース消費者 | 基本的には Solaris のプロセス。プロジェクトやタスクなどのプロセスモデルエンティティにより、集計済みのリソース消費に関して考察できます。 |
| リソース制御 | プロセスごと、タスクごと、またはプロジェクトごとのリソース消費量に対する制限。 |
| リソースセット | プロセスをバインド可能なリソース。たいていの場合、カーネルサブシステムにより構築され、ある種の区分化を提供するオブジェクトを指して使用されます。リソースセットの例には、スケジューリングクラスやプロセッサセットが含まれます。 |
| リソースパーティション | リソースの排他的な一部。リソースのパーティションすべての合計は、実行中の単一の Solaris インスタンスで利用可能なリソースの総量を表します。 |
| リソースプール | マシンのリソース区分化に使用する構成メカニズム。リソースプールは、区分化可能なリソースグループ間の関係を表します。 |
| ロックされたメモリー | ページング不可能なメモリー。 |
| cap | システムリソース使用率に対する規制。 |
| CMT リソース | CPUS、コア、およびソケット。 |

| | |
|----------------------------------|--|
| CPU | ゾーンのコンテキストでは、ハードウェアスレッドを指します。 |
| FSS | 公平配分スケジューラ を参照してください。 |
| Oracle Solaris 10 ゾーン | Oracle Solaris 11 リリースを実行しているシステムの <code>solaris10</code> ブランドゾーンで実行される Solaris 10 アプリケーション用の完全な実行時環境を提供するソフトウェアパーティション技術。 |
| Oracle Solaris カーネルゾーン | ゾーン内の完全なカーネルおよびユーザー環境を提供し、さらにホストとゾーンの間のカネルの分離を強化するソフトウェアパーティション技術。 |
| Oracle Solaris ゾーン | オペレーティングシステムサービスを仮想化し、アプリケーションを実行するための分離およびセキュリティ保護された環境を提供するソフトウェア区分技術。 |
| RSS | 常駐セットサイズ を参照してください。 |
| WSS | 作業セットサイズ を参照してください。 |

索引

あ

アプリケーションと capped-cpu, 38
一時プール, 37

か

機能

排他的 IP ゾーン, 47
共有 IP ゾーン, 46
構成可能な特権、ゾーン, 54
公平配分スケジューラ (FSS), 39

さ

スワップ領域の上限, 40
ゾーン

anet, 65, 71
bootargs プロパティ, 64
capped-cpu, 65
capped-memory, 40, 65
dedicated-cpu, 64
ip-type, 64
IPoIB (solaris のみ), 71
limitpriv, 64
net, 65
Oracle Solaris の制限と機能, 28
pool, 67
rootzpool, 68
scheduling-class, 64
virtual-cpu, 65
機能, 26
共有 IP, 46
権限、役割、プロファイル, 34
構成, 58

構成可能な特権, 54
構成の概要, 35
作成, 21
種類別の特性, 19
状態, 21
状態モデル, 21
ゾーン規模のリソース制御, 62
ディスク形式のサポート, 53
定義, 10
デフォルト以外の, 15
データセット, 65
排他的 IP, 47
ブランド, 15
プロパティタイプ, 62
モニタリング, 26
ライブ再構成, 31
リソース制御, 55
リソースタイプ, 62
リソースタイプのプロパティ, 68
ゾーン管理者, 20
ゾーン規模のリソース制御, 55
ゾーンの admin 承認, 36
ゾーン名, 19
ゾーン ID, 19

た

大域管理者, 18, 20
大域ゾーン, 18
ディスク形式のサポート
ゾーン, 53
データリンク, 45
デバイスリソース
ストレージ URI, 53
デフォルト以外の
ゾーン, 15

は

- 排他的 IP ゾーン, 47
- 非大域ゾーン, 18
- 非大域ゾーン管理者, 18
- 物理メモリの上限, 40
- 不変ゾーン
 - 読み取り専用ゾーン, 11
- ブランド, 12, 13
- ブランドゾーン, 15
 - 実行中のプロセス, 16

や

- 読み取り専用ゾーン
 - file-mac-profile, 36
- 読み取り専用ゾーンルート, 36

ら

- ライブゾーン再構成, 31
- リソース制御
 - ゾーン規模, 55
- リムーバブル lofi デバイス, 52
- ロックされたメモリの上限, 40

A

- allowed-addresses
 - 排他的 IP ゾーン, 47
- autoboot, 36

B

- bootargs プロパティ, 64
- BrandZ, 15

C

- capped-cpu リソース, 38, 65
- capped-memory, 65
- capped-memory リソース, 40

D

- dedicated-cpu リソース, 37, 64
- defrouter, 75
 - 排他的 IP ゾーン, 47
- DHCP
 - 排他的 IP ゾーン, 47
- dtrace_proc, 64
- dtrace_user, 64

E

- EVS
 - ゾーン, 45

H

- hostid, 52

I

- ip-type プロパティ, 64
- IP フィルタ
 - 排他的 IP ゾーン, 47
- IP ルーティング
 - 排他的 IP ゾーン, 47
- ipkg ゾーン
 - solaris へのマップ, 28
 - 変換, 12
- IPMP
 - 排他的 IP ゾーン, 47
- IPoIB, 74

L

- limitpriv プロパティ, 64
- linkmode, 73
- lofi デバイス
 - リムーバブル, 52

N

- net リソース
 - 共有 IP ゾーン, 46

排他的 IP ゾーン, 47

O

Oracle Solaris カーネルゾーン, 12

Oracle Solaris クラスタ
ゾーンクラスタ, 14

Oracle Solaris ゾーン, 13

P

pkey, 72, 74

pkg update 中の autoboot の無効化, 36

pool プロパティ, 67

R

RDS (Reliable Datagram Sockets), 49

rootzpool リソース

solaris ブランド, 41

S

scheduling-class プロパティ, 64

SMF サービス

大域ゾーン, 25

非大域ゾーン, 26

solaris, 13

solaris 非大域ゾーン

Oracle Solaris, 28

V

virtual-cpu リソース, 38, 65

Z

ZFS

データセット, 65

zone.cpu-cap リソース制御, 55

zone.cpu-shares リソース制御, 55

zone.max-locked-memory リソース制御, 55

zone.max-lofi リソース制御, 55

zone.max-lwps リソース制御, 55

zone.max-msg-ids リソース制御, 56

zone.max-processes リソース制御, 56

zone.max-sem-ids リソース制御, 56

zone.max-shm-ids リソース制御, 56

zone.max-shm-memory リソース制御, 56

zone.max-swap リソース制御, 56

zonecfg

admin 承認, 36

template, 34

一時プール, 37

エンティティ, 62

サブコマンド, 59

操作, 35

大域ゾーン内, 58

モード, 59

有効範囲, 59

有効範囲、大域, 59

有効範囲、リソース固有, 59

zpool リソース, 43

