

Oracle® VM Server for SPARC 3.3 管理ガイド



Part No: E64643
2015 年 10 月

Part No: E64643

Copyright © 2007, 2015, 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、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.

このソフトウェアまた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。このソフトウェアまたはハードウェアは、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。このソフトウェアまた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際、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。このソフトウェアまた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、Oracle Corporation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はオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

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。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporation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。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporation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。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

このドキュメントの使用方法	11
I Oracle VM Server for SPARC 3.3 ソフトウェア	13
1 Oracle VM Server for SPARC ソフトウェアの概要	15
Oracle VM Server for SPARC と Oracle Solaris OS のバージョンについて	15
ハイパーバイザと Logical Domains	16
Logical Domains Manager	18
Oracle VM Server for SPARC 物理から仮想への変換ツール	22
Oracle VM Server for SPARC 管理情報ベース	23
Oracle VM Server for SPARC のトラブルシューティング	23
2 Oracle VM Server for SPARC のセキュリティー	25
権利の使用による論理ドメインの管理の委任	25
権利の使用によるドメインコンソールへのアクセスの制御	30
Logical Domains Manager の監査の有効化および使用	36
ドメインコンソールのロギングの使用	38
3 サービスおよび制御ドメインの設定	41
出力メッセージ	41
デフォルトのサービスの作成	42
制御ドメインの初期構成	43
ドメインを使用するためのリブート	46
Oracle Solaris 10 サービスドメインとその他のドメインの間のネットワークの有効化	47
仮想ネットワーク端末サーバーデーモンの有効化	48
ILOM 相互接続が有効になっていることの確認	49
4 ゲストドメインの設定	51
ゲストドメインの作成と起動	51
ゲストドメインへの Oracle Solaris OS のインストール	55
5 I/O ドメインの構成	61

I/O ドメインの概要	61
I/O ドメインを作成するための一般的なガイドライン	62
6 PCIe バスの割り当てによってルートドメインを作成する方法	63
PCIe バスの割り当てによってルートドメインを作成する方法	63
7 PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成	73
SR-IOV の概要	73
SR-IOV のハードウェア要件とソフトウェア要件	76
SR-IOV 機能の現在の制限事項	80
静的 SR-IOV	81
動的 SR-IOV	83
I/O 仮想化の有効化	84
PCIe SR-IOV 仮想機能の使用の計画	85
Ethernet SR-IOV 仮想機能の使用	87
InfiniBand SR-IOV 仮想機能の使用	107
ファイバチャネル SR-IOV 仮想機能の使用	121
I/O ドメインの回復性	136
非信頼性 I/O ドメインが構成された状態のルートドメインのリブート	144
8 直接 I/O を使用した I/O ドメインの作成	147
Creating an I/O Domain by Assigning PCIe Endpoint Devices	147
直接 I/O のハードウェア要件とソフトウェア要件	150
直接 I/O 機能の現在の制限事項	151
PCIe エンドポイントデバイス構成の計画	152
PCIe エンドポイントを構成した状態のルートドメインのリブート	154
PCIe ハードウェアの変更	156
PCIe エンドポイントデバイスを割り当てることによる I/O ドメインの作成	159
9 primary 以外のルートドメインの使用	167
primary 以外のルートドメインの概要	167
primary 以外のルートドメインの要件	169
primary 以外のルートドメインの制限	169
primary 以外のルートドメインの例	170
10 仮想ディスクの使用	177
仮想ディスクの概要	177
仮想ディスクの識別子とデバイス名	179
仮想ディスクの管理	180
仮想ディスクの表示	183
仮想ディスクバックエンドオプション	184
仮想ディスクバックエンド	186

仮想ディスクマルチパスの構成	194
CD、DVD および ISO イメージ	200
仮想ディスクのタイムアウト	203
仮想ディスクおよび SCSI	204
仮想ディスクおよび format コマンド	205
仮想ディスクと ZFS の使用	205
Oracle VM Server for SPARC 環境でのボリュームマネージャーの使用	210
仮想ディスクに関する問題	214
11 仮想 SCSI ホストバスアダプタの使用	219
仮想 SCSI ホストバスアダプタの概要	219
仮想 SCSI HBA の動作モデル	222
仮想 SCSI HBA の識別子とデバイス名	223
仮想 SCSI HBA の管理	224
ゲストドメイン内の仮想 LUN の見え方	229
仮想 SCSI HBA と仮想 SAN の構成	229
仮想 SCSI HBA マルチパスの構成	230
SCSI デバイスからのブート	234
仮想 LUN のインストール	235
仮想 SCSI HBA のタイムアウト	236
仮想 SCSI HBA および SCSI	236
ゲストドメインでの断片化の激しい I/O バッファのサポート	237
12 仮想ネットワークの使用	239
仮想ネットワークの概要	240
Oracle Solaris 11 ネットワークの概要	240
Oracle Solaris 10 ネットワークの概要	243
仮想ネットワークパフォーマンスの最大化	245
仮想スイッチ	247
仮想ネットワークデバイス	248
ネットワークデバイス構成および統計情報の表示	251
仮想ネットワークデバイスで消費される物理ネットワーク帯域幅の量の制 御	254
仮想デバイス識別子およびネットワークインタフェース名	258
自動または手動による MAC アドレスの割り当て	261
Oracle Solaris 10 を実行するドメインでのネットワークアダプタの使用	264
NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成	265
Oracle VM Server for SPARC 環境での IPMP の構成	269
VLAN のタグ付けの使用	279
プライベート VLAN の使用	284
パケットスループットパフォーマンスの調整	290

NIU ハイブリッド I/O の使用	291
仮想スイッチでのリンクアグリゲーションの使用	296
ジャンボフレームの構成	299
Oracle Solaris 11 のネットワーク固有の機能の相違点	304
仮想ネットワーク上の仮想 NIC の使用	306
13 ドメインの移行	311
ドメインの移行の概要	312
移行処理の概要	312
ソフトウェアの互換性	313
移行処理のセキュリティー	313
ドメイン移行の FIPS 140-2 モード	315
ドメイン移行の制限	318
ドメインの移行	321
アクティブなドメインの移行	322
バインドされたドメインまたはアクティブでないドメインの移行	330
進行中の移行のモニタリング	331
進行中の移行の取り消し	332
移行の失敗からの回復	332
移行の例	333
14 リソースの管理	337
リソースの再構成	337
リソースの割り当て	339
CPU の割り当て	340
ハードパーティションによるシステムの構成	344
ドメインへの物理リソースの割り当て	351
メモリーの動的再構成の使用	356
リソースグループの使用	364
電源管理の使用	365
動的なリソース管理の使用	366
ドメインリソースの一覧表示	369
パフォーマンスカウンタプロパティの使用	376
15 ドメイン構成の管理	381
ドメイン構成の管理	381
使用可能な構成回復方法	382
サービスプロセッサの接続の問題への対処	389
16 ハードウェアエラーの処理	391
ハードウェアエラー処理の概要	391
FMA を使用した障害のあるリソースのブラックリスト登録または構成解除 ...	392

障害のあるリソースまたは見つからないリソースを検出したあとのドメイン復旧	393
ドメインの縮退化	397
I/O リソースを退避としてマーク	397
17 その他の管理タスクの実行	399
CLI での名前の入力	400
/etc/system ファイルのプロパティ値の更新	400
ネットワークを介したゲストコンソールへの接続	401
コンソールグループの使用	401
負荷が大きいドメインの停止処理がタイムアウトする可能性	402
Oracle VM Server for SPARC による Oracle Solaris OS の運用	403
サービスプロセッサでの Oracle VM Server for SPARC の使用	405
ドメインの依存関係の構成	406
CPU およびメモリアドレスのマッピングによるエラー発生箇所の確認	410
ユニバーサル固有識別子の使用	413
仮想ドメイン情報コマンドと API	413
論理ドメインチャネルの使用	414
多数のドメインのブート	418
Oracle VM Server for SPARC システムの正常な停止と電源の再投入	419
Logical Domains 変数の永続性	420
割り込み制限の調整	421
ドメイン I/O の依存関係の一覧表示	424
Logical Domains Manager デーモンの有効化	425
自動保存構成データの保存および復元	426
出荷時デフォルト構成とドメインの無効化	428
II オプションの Oracle VM Server for SPARC ソフトウェア	431
18 Oracle VM Server for SPARC テンプレートの使用	433
Oracle VM Server for SPARC テンプレートの概要	433
Oracle VM Server for SPARC テンプレートユーティリティのインストール	434
Oracle VM Server for SPARC テンプレートのライフサイクル	434
Oracle VM Server for SPARC テンプレートの例	438
19 Oracle VM Server for SPARC 物理から仮想への変換ツール	447
Oracle VM Server for SPARC P2V ツールの概要	447
バックエンドデバイス	450
Oracle VM Server for SPARC P2V ツールのインストールおよび構成	451

ldmp2v コマンドの使用	454
Oracle VM Server for SPARC 物理から仮想への変換ツールの既知の問題	463
20 電源管理の使用	467
電源管理の使用	467
21 Oracle VM Server for SPARC 管理情報ベース (MIB) ソフトウェアの使用	475
Oracle VM Server for SPARC 管理情報ベースの概要	475
Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成	480
セキュリティの管理	483
ドメインのモニタリング	484
SNMP トラップの使用	508
ドメインの起動と停止	515
22 Logical Domains Manager の検出	521
Logical Domains Manager を実行しているシステムの検出	521
23 Logical Domains Manager での XML インタフェースの使用	525
XML トランスポート	525
XML プロトコル	526
イベントメッセージ	532
Logical Domains Manager のアクション	538
Logical Domains Manager のリソースとプロパティ	540
XML スキーマ	559
用語集	561
索引	571

このドキュメントの使用方法

- **概要** – サポートされるサーバー、ブレード、およびサーバーモジュールでの Oracle VM Server for SPARC 3.3 ソフトウェアの概要、セキュリティ上の考慮事項、インストール、構成、変更、および一般的なタスクの実行に関する詳細な情報や手順について説明します。『Oracle VM Server for SPARC 3.3 インストールガイド』の「サポートされるプラットフォーム」を参照してください。

注記 - 本書で説明する機能は、『Oracle VM Server for SPARC 3.3 インストールガイド』に記載されているすべてのサポート対象のソフトウェアおよびハードウェアプラットフォームで使用できます。ただし、一部の機能は、サポートされているシステムソフトウェアおよびハードウェアプラットフォームのサブセット上でしか使用できません。このような例外については、『Oracle VM Server for SPARC 3.3 リリースノート』の「このリリースの最新情報」および [What's New in Oracle VM Server for SPARC Software \(http://www.oracle.com/technetwork/server-storage/vm/documentation/sparc-whatsnew-330281.html\)](http://www.oracle.com/technetwork/server-storage/vm/documentation/sparc-whatsnew-330281.html) を参照してください。

- **対象読者** – SPARC サーバー上の仮想化を管理するシステム管理者
- **必要な知識** – これらのサーバーのシステム管理者は、UNIX システムおよび Oracle Solaris オペレーティングシステム (Oracle Solaris OS) の実践的な知識を持っている必要があります

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html> で入手可能です。

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。

パート I

Oracle VM Server for SPARC 3.3 ソフトウェア

ここでは、SPARC T シリーズ、SPARC M シリーズ、および Fujitsu M10 プラットフォームに高い効率性とエンタープライズクラスの仮想化機能を提供する Oracle VM Server for SPARC 3.3 ソフトウェアを紹介します。

- 第1章「Oracle VM Server for SPARC ソフトウェアの概要」
- 第2章「Oracle VM Server for SPARC のセキュリティー」
- 第3章「サービスおよび制御ドメインの設定」
- 第4章「ゲストドメインの設定」
- 第5章「I/O ドメインの構成」
- 第6章「PCIe バスの割り当てによってルートドメインを作成する方法」
- 第7章「PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成」
- 第8章「直接 I/O を使用した I/O ドメインの作成」
- 第9章「primary 以外のルートドメインの使用」
- 第10章「仮想ディスクの使用」
- 第11章「仮想 SCSI ホストバスアダプタの使用」
- 第12章「仮想ネットワークの使用」
- 第13章「ドメインの移行」
- 第14章「リソースの管理」
- 第15章「ドメイン構成の管理」

-
- 第16章「ハードウェアエラーの処理」
 - 第17章「その他の管理タスクの実行」

◆◆◆ 第 1 章

Oracle VM Server for SPARC ソフトウェアの概要

この章では、Oracle VM Server for SPARC ソフトウェアの概要を紹介します。

Oracle VM Server for SPARC は、Oracle SPARC T シリーズおよび SPARC M シリーズプラットフォーム、および Fujitsu M10 プラットフォームに効率性の高い、エンタープライズクラスの仮想化機能を提供します。Oracle VM Server for SPARC ソフトウェアを使用すると、最大で 128 台の仮想サーバーを単一のシステム上に作成できます。これは論理ドメインと呼ばれます。こうした構成により、SPARC T シリーズおよび SPARC M シリーズプラットフォーム、および Fujitsu M10 プラットフォーム および Oracle Solaris OS が提供する大規模なスレッドを活用できるようになります。

この章では、次の項目について説明します。

- 15 ページの「Oracle VM Server for SPARC と Oracle Solaris OS のバージョンについて」
- 16 ページの「ハイパーバイザと Logical Domains」
- 18 ページの「Logical Domains Manager」
- 22 ページの「Oracle VM Server for SPARC 物理から仮想への変換ツール」
- 23 ページの「Oracle VM Server for SPARC 管理情報ベース」
- 23 ページの「Oracle VM Server for SPARC のトラブルシューティング」

Oracle VM Server for SPARC と Oracle Solaris OS のバージョンについて

Oracle VM Server for SPARC ソフトウェアを使用するには、Oracle Solaris OS の特定のバージョンが使用され、必要なソフトウェアパッチが適用され、特定のバージョンのシステムファームウェアが存在することが必要です。詳細は、『[Oracle VM Server for SPARC 3.3 インストールガイド](#)』の「完全に認定された Oracle Solaris OS バージョン」を参照してください。

ゲストドメインで動作するバージョンの Oracle Solaris OS は、primary ドメインで動作する Oracle Solaris OS バージョンから独立しています。そのため、primary ドメインで Oracle Solaris 11 OS を実行している場合でも、ゲストドメインでは引き続き Oracle Solaris 10 OS を実行できます。

注記 - 一部のプラットフォームでは primary ドメインで Oracle Solaris 10 OS がサポートされなくなりました。使用しているプラットフォームのマニュアルを確認してください。その他のドメインでは引き続き Oracle Solaris 10 OS を実行できます。

primary ドメインでの Oracle Solaris 10 OS と Oracle Solaris 11 OS の実行の違いは、それぞれの OS の機能の違いだけです。

ハイパーバイザと Logical Domains

このセクションでは、論理ドメインをサポートしている SPARC ハイパーバイザの概要を説明します。

SPARC ハイパーバイザは、小さなファームウェア層で、オペレーティングシステムを記述できる安定した仮想化マシンアーキテクチャーを提供します。ハイパーバイザを使用する SPARC サーバーでは、論理オペレーティングシステムの動作をハイパーバイザが制御できるようにするためのハードウェア機能が用意されています。

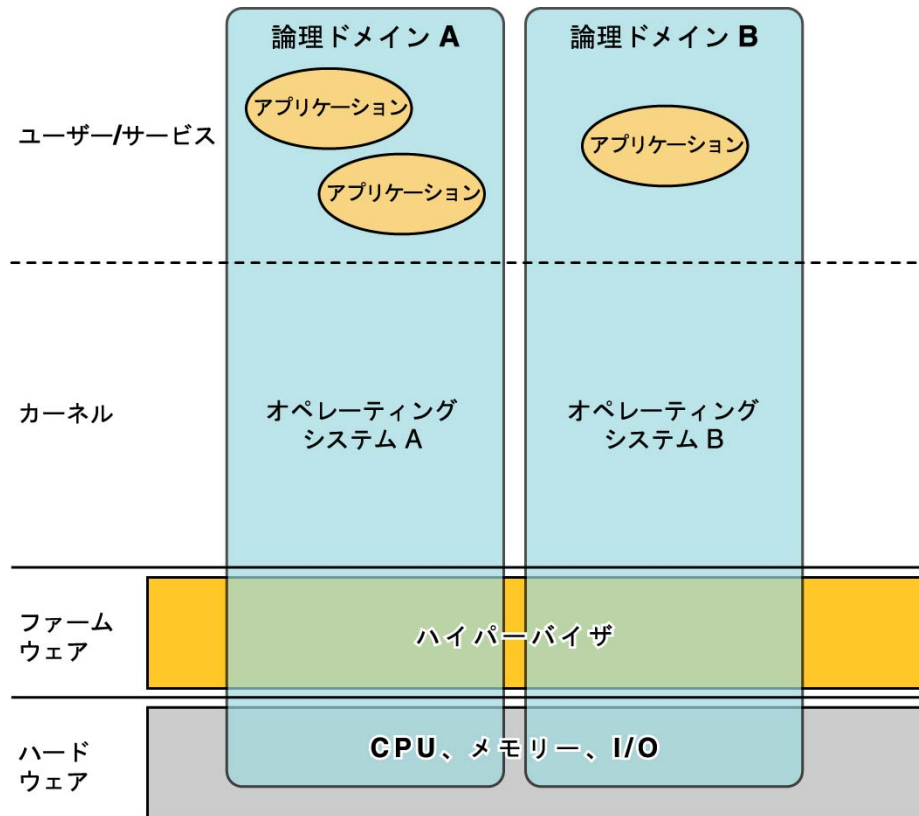
論理ドメインは、リソースの個別の論理グループで構成される仮想マシンです。論理ドメインは、単一のコンピュータシステム内で独自のオペレーティングシステムおよび ID を持っています。各論理ドメインは、サーバーの電源の再投入を実行する必要なしに、作成、削除、再構成、およびリブートを単独で行うことができます。異なる論理ドメインでさまざまなアプリケーションソフトウェアを実行できます。また、パフォーマンスおよび安全性の目的から、これらを独立した状態にしておくことができます。

各論理ドメインは、ハイパーバイザがそのドメインに対して利用可能にしたサーバーリソースに対してのみ、監視および対話が許可されています。Logical Domains Manager により、制御ドメイン経由でハイパーバイザが行う動作を指定できます。つまり、ハイパーバイザは、サーバーのリソースをパーティションに分割し、限定的なサブセットを複数のオペレーティングシステム環境に提供します。このパーティションの分割と提供は、論理ドメインを作成する場合の基本的なメカニズムです。次の図に、2 つの論理ドメインをサポートするハイパーバイザを示します。また、Oracle VM Server for SPARC の機能を構成する次の層についても示します。

- ユーザー/サービス (アプリケーション)
- カーネル (オペレーティングシステム)

- ファームウェア (ハイパーバイザ)
- ハードウェア (CPU、メモリー、I/O など)

図 1-1 2つのドメインをサポートするハイパーバイザ



特定の SPARC ハイパーバイザがサポートする各論理ドメインの数と機能は、サーバーによって異なります。ハイパーバイザは、サーバー全体の CPU、メモリー、および I/O リソースのサブセットを特定の論理ドメインに割り当てることができます。この機能により、それぞれが独自の論理ドメイン内にある複数のオペレーティングシステムを同時にサポートすることができます。リソースは、任意に細分化して個々の論理ドメイン間で再配置できます。たとえば、CPU は CPU スレッド単位で論理ドメインに割り当てることができます。

各論理ドメインは、次のような独自のリソースを使用して、完全に独立したマシンとして管理できます。

- カーネル、パッチ、およびチューニングパラメータ

- ユーザーアカウントおよび管理者
- ディスク
- ネットワークインタフェース、MAC アドレス、および IP アドレス

各論理ドメインは、サーバーの電源再投入を実行しなくても、互いに独立して停止、起動、およびリブートできます。

ハイパーバイザソフトウェアは、論理ドメイン間の分離を維持する役割を果たします。ハイパーバイザソフトウェアは、論理ドメインが相互に通信できるように論理ドメインチャネル (LDC) も提供します。LDC を使用することで、ドメインはネットワークサービスやディスクサービスなどのサービスを相互に提供できます。

サービスプロセッサ (SP) はシステムコントローラ (SC) と呼ばれ、物理マシンのモニタリングと実行を行います。論理ドメインの管理は行いません。Logical Domains Manager が、論理ドメインの管理を行います。

Oracle VM Server for SPARC ソフトウェアの管理には、`ldm` コマンドに加えて、Oracle VM Manager も使用できるようになりました。

Oracle VM Manager は、Oracle VM 環境を管理する際に使用可能な Web ベースのユーザーインタフェースです。このユーザーインタフェースの以前のバージョンでは Oracle VM Server x86 ソフトウェアしか管理されませんでしたが、Oracle VM Manager 3.2 および Oracle VM Server for SPARC 3.0 以降のバージョンでは、Oracle VM Server for SPARC ソフトウェアも管理できます。Oracle VM Manager の詳細は、[Oracle VM Documentation \(http://www.oracle.com/technetwork/documentation/vm-096300.html\)](http://www.oracle.com/technetwork/documentation/vm-096300.html) を参照してください。

Logical Domains Manager

Logical Domains Manager は、論理ドメインの作成と管理、および論理ドメインの物理リソースへのマッピングに使用されます。サーバーで実行できる Logical Domains Manager は 1 つだけです。

ドメインの役割

論理ドメインはすべて同じですが、論理ドメインに対して指定する役割に基づいてそれぞれ区別できます。論理ドメインが実行できる役割は、次のとおりです。

- **制御ドメイン**。Logical Domains Manager がこのドメインで実行されることで、他のドメインを作成して管理し、仮想リソースを他のドメインに割り当てることができます。制御ドメインは、サーバーごとに 1 つだけ存在できます。制御ドメインは、Oracle VM Server for SPARC ソフトウェアをインストールするときに最初に作成されるドメインです。制御ドメインには、`primary` という名前が付けられます。
- **サービスドメイン**。サービスドメインは、仮想スイッチ、仮想コンソール端末集配信装置、仮想ディスクサーバーなどの仮想デバイスサービスをほかのドメインへ提供します。複数のサービスドメインを保持でき、どのドメインもサービスドメインとして構成できます。
- **I/O ドメイン**。I/O ドメインは、PCI EXPRESS (PCIe) コントローラのネットワークカードなどの物理 I/O デバイスに直接アクセスできます。I/O ドメインは、次を所有できます。
 - PCIe ルートコンプレックス。
 - PCIe スロット、または直接 I/O (DIO) 機能を使用することによるオンボード PCIe デバイス。[147 ページの「Creating an I/O Domain by Assigning PCIe Endpoint Devices」](#)を参照してください。
 - PCIe SR-IOV 仮想機能。[第7章「PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成」](#)を参照してください。

I/O ドメインは、I/O ドメインがサービスドメインとしても使用される場合に、仮想デバイスの形式でほかのドメインと物理 I/O デバイスを共有できます。

- **ルートドメイン**。ルートドメインには PCIe ルートコンプレックスが割り当てられています。このドメインは PCIe ファブリックを所有し、ファブリックのエラー処理など、ファブリックに関連するすべてのサービスを提供します。ルートドメインは I/O ドメインでもあり、物理 I/O デバイスを所有し、それらに直接アクセスできます。

保持できるルートドメインの数は、プラットフォームアーキテクチャーによって決まります。たとえば、8 ソケットの Oracle SPARC T5-8 サーバーを使用している場合は、最大 16 のルートドメインを保持できます。

デフォルトのルートドメインは、`primary` ドメインです。`primary` 以外のドメインを使用して、ルートドメインとして機能させることができます。

- **ゲストドメイン**。ゲストドメインは I/O を行わないドメインで、1 つ以上のサービスドメインにより提供される仮想デバイスサービスを使用します。ゲストドメインは、物理 I/O デバイスを

持っておらず、仮想ディスクや仮想ネットワークインタフェースなどの仮想 I/O デバイスのみを持ちます。

まだ Oracle VM Server for SPARC が構成されていない既存のシステムに、Logical Domains Manager をインストールできます。この場合、OS の現在のインスタンスが制御ドメインになります。また、システムは 1 つのドメインである、制御ドメインによってのみ構成されます。制御ドメインを構成したあと、ドメインを追加してアプリケーションを制御ドメインから新しいドメインに移動することによって、システム全体をもっとも効率的に利用できるように、アプリケーションの負荷をほかのドメイン間で分散できます。

コマンド行インタフェース

Logical Domains Manager は、コマンド行インタフェース (Command-Line Interface、CLI) を使用して論理ドメインの作成と構成を行います。CLI には、単一のコマンド `ldm` があり、これは複数のサブコマンドを備えています。[ldm\(1M\)](#) のマニュアルページを参照してください。

Logical Domains Manager CLI を使用するには、Logical Domains Manager デモン `ldmd` が実行されている必要があります。

仮想入出力

Oracle VM Server for SPARC 環境では、システムで最大 128 個 (Fujitsu M10 サーバーで最大 256 個) のドメインをプロビジョニングできます。一部のサーバー、特に単一プロセッサと一部のデュアルプロセッサシステムでは、I/O バスおよび物理 I/O スロットの数に制限があります。そのため、これらのシステムのすべてのドメインに対して、物理ディスクおよびネットワークデバイスへの排他的なアクセスを提供できるとは限りません。物理デバイスへのアクセスを提供するため、ドメインに PCIe バスやエンドポイントデバイスを割り当てることができます。この解決方法は、すべてのドメインにデバイスへの排他的なアクセスを提供するには不十分です。直接アクセス可能な物理 I/O デバイス数の制限には、仮想 I/O モデルの実装により対処できます。[第5章「I/O ドメインの構成」](#)を参照してください。

物理 I/O アクセスを行わない論理ドメインは、サービスドメインと通信する仮想 I/O デバイスを使用して構成されます。サービスドメインは、仮想デバイスサービスを実行して、物理デバイスまたはその機能にアクセスを提供します。このようなクライアントサーバーモデルで、仮想 I/O デバイスは、論理ドメインチャネル (LDC) と呼ばれるドメイン間通信チャネルを使用して、相互

に、またはサービスの対象と通信します。仮想化 I/O 機能には、仮想ネットワーク、ストレージ、およびコンソールのサポートが含まれています。

仮想ネットワーク

Oracle VM Server for SPARC は、仮想ネットワークデバイスと仮想ネットワークスイッチデバイスを使用して、仮想ネットワークを実装します。仮想ネットワーク (vnet) デバイスは、Ethernet デバイスをエミュレートし、ポイントツーポイントチャネルを使用してシステム内のほかの vnet デバイスと通信します。仮想スイッチ (vsw) デバイスは、主に仮想ネットワークのすべての受信パケットおよび送信パケットのマルチプレクサとして機能します。vsw デバイスは、サービスドメインの物理ネットワークアダプタに直接接続し、仮想ネットワークの代わりにパケットを送受信します。vsw デバイスは、単純なレイヤー 2 スイッチとしても機能し、システム内で vsw デバイスに接続された vnet デバイス間でパケットをスイッチします。

仮想ストレージ

仮想ストレージインフラストラクチャーは、クライアントサーバーモデルを使用して、論理ドメインに直接割り当てられていないブロックレベルのストレージに論理ドメインがアクセスできるようにします。このモデルは、次のコンポーネントを使用します。

- ブロック型デバイスインタフェースをエクスポートする仮想ディスククライアント (vdc)
- 仮想ディスクサービス (vds) は、仮想ディスククライアントの代わりにディスク要求を処理し、サービスドメインに存在するバックエンドのストレージへ発行します

クライアントドメインでは仮想ディスクは通常のディスクとして認識されますが、ほとんどのディスク操作は仮想ディスクサービスに転送され、サービスドメインで処理されます。

仮想コンソール

Oracle VM Server for SPARC 環境では、primary ドメインからのコンソール I/O はサービスプロセッサに送信されます。ほかのすべてのドメインからのコンソール I/O は、仮想コンソール端末集配信装置 (vcc) を実行しているサービスドメインにリダイレクトされます。通常、vcc を実行するドメインは、primary ドメインです。仮想コンソール端末集配信装置サービスは、すべてのドメインのコンソールトラフィックの端末集配信装置として機能します。また、仮想ネットワーク端末サーバーデーモン (vntsd) とのインタフェースを提供し、UNIX ソケットを使用して各コンソールへのアクセスを提供します。

リソースの構成

Oracle VM Server for SPARC ソフトウェアを実行するシステムは、仮想 CPU、仮想 I/O デバイス、暗号化ユニット、メモリーなどのリソースを構成できます。一部のリソースは実行中のドメインで動的に構成可能ですが、他のリソースは停止中のドメインで構成する必要があります。制御ドメインでリソースを動的に構成できない場合は、まず遅延再構成を開始する必要があります。遅延再構成は、制御ドメインのリブートが完了するまで構成処理を延期します。詳細については、[337 ページの「リソースの再構成」](#)を参照してください。

持続的な構成

ldm コマンドを使用して、論理ドメインの現在の構成をサービスプロセッサに格納できます。構成の追加、使用する構成の指定、構成の削除、および構成の表示を行うことができます。詳細については、[ldm\(1M\)](#) のマニュアルページを参照してください。[405 ページの「サービスプロセッサでの Oracle VM Server for SPARC の使用」](#)で説明するように、SP からブートするように構成を指定することもできます。

構成の管理については、[381 ページの「ドメイン構成の管理」](#)を参照してください。

Oracle VM Server for SPARC 物理から仮想への変換ツール

Oracle VM Server for SPARC Physical-to-Virtual (P2V) 変換ツールは、既存の物理システムを、チップマルチスレッディング (CMT) システム上の論理ドメインで Oracle Solaris 10 OS を実行する仮想システムに自動的に変換します。Oracle Solaris 11 OS を実行している制御ドメインから ldmp2v コマンドを実行して、次のソースシステムのいずれかを論理ドメインに変換できます。

- Solaris 8、Solaris 9、または Oracle Solaris 10 OS を実行している、任意の sun4u SPARC ベースのシステム
- Oracle Solaris 10 OS を実行しているが、Oracle VM Server for SPARC ソフトウェアは実行していない任意の sun4v システム

注記 - Oracle Solaris 11 物理システムを仮想システムに変換するために、P2V ツールを使用することはできません。

ツールおよびそのインストールについては、[第19章「Oracle VM Server for SPARC 物理から仮想への変換ツール」](#)を参照してください。ldmp2v コマンドについては、[ldmp2v\(1M\)](#) マニュアルページを参照してください。

Oracle VM Server for SPARC 管理情報ベース

Oracle VM Server for SPARC 管理情報ベース (Management Information Base、MIB) を使用すると、サードパーティーのシステム管理アプリケーションによるドメインのリモートモニタリングと、簡易ネットワーク管理プロトコル (SNMP) による論理ドメイン (ドメイン) の起動および停止が可能になります。詳細については、[第21章「Oracle VM Server for SPARC 管理情報ベース \(MIB\) ソフトウェアの使用」](#)を参照してください。

Oracle VM Server for SPARC のトラブルシューティング

Oracle VM Server for SPARC ソフトウェアの特定の問題に関する情報は、次の資料から入手できます。

- 『Oracle VM Server for SPARC 3.3 リリースノート』の「既知の問題」
- Information Center: Overview of Oracle VM Server for SPARC (LDoms) (Doc ID 1589473.2) (https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay?_afrcLoop=227880986952919&id=1589473.2&_afrcWindowMode=0&_adf.ctrl-state=wu098o5r6_96)

Oracle VM Server for SPARC のセキュリティー

この章では、Oracle VM Server for SPARC システムで有効にできるいくつかのセキュリティー機能について説明します。

この章では、次の項目について説明します。

- 25 ページの「権利の使用による論理ドメインの管理の委任」
- 30 ページの「権利の使用によるドメインコンソールへのアクセスの制御」
- 36 ページの「Logical Domains Manager の監査の有効化および使用」
- 38 ページの「ドメインコンソールのログインの使用」

注記 - このマニュアルの例は、スーパーユーザーが実行していることを示しています。ただし、代わりにプロファイルを使用すれば、ユーザーが管理タスクを実行するためのより詳細なアクセス権を取得できるようになります。

権利の使用による論理ドメインの管理の委任

Logical Domains Manager パッケージは、2 つの定義済みの権利プロファイルを、ローカルの権利構成に追加します。これらの権利プロファイルは、特権のないユーザーに管理特権を委任します。

- LDoms Management プロファイルは、ユーザーにすべての `ldm` サブコマンドの使用を許可します。
- LDoms Review プロファイルは、ユーザーにすべてのリスト関連の `ldm` サブコマンドの使用を許可します。

これらの権利プロファイルは、ユーザー、またはその後ユーザーに割り当てられる役割に、直接割り当てることができます。これらのプロファイルのいずれかがユーザーに直接割り当てられている場合、ドメインを管理するために `ldm` コマンドを正常に使用するには、`pfexec` コマンド、または `pfbash` または `pfksh` などのプロファイルシェルを使用する必要があります。役割または権利プロファイルのどちらを使用するかは、使用している権利構成に基づいて決定します。『[System](#)

[Administration Guide: Security Services](#)』または『[Securing Users and Processes in Oracle Solaris 11.3](#)』を参照してください。

ユーザー、承認、権利プロファイル、および役割は、次の方法で構成できます。

- ファイルを使用してシステム上でローカルで構成する
- LDAP などのネームサービスで一元的に構成する

Logical Domains Manager をインストールすると、必要な権利プロファイルがローカルファイルに追加されます。ネームサービスでプロファイルおよび役割を構成するには、『[System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)』を参照してください。Logical Domains Manager パッケージによって提供される承認および実行属性の概要は、[29 ページ](#)の「[Logical Domains Manager プロファイルの内容](#)」を参照してください。この章の例はすべて、権利構成がローカルファイルを使用すると仮定しています。

権利プロファイルと役割の使用



注意 - `usermod` および `rolemod` コマンドを使用して承認、権利プロファイル、または役割を追加する際は、注意してください。

- Oracle Solaris 11 OS の場合は、追加する承認ごとにプラス記号 (+) を使用して値を追加します。

たとえば、`usermod -A +auth username` コマンドは、`rolemod` コマンドと同様に、`auth` 承認を `username` ユーザーに付与します。

- Oracle Solaris 10 OS の場合、`usermod` または `rolemod` コマンドは既存の値を置き換えます。

値を置き換える代わりに追加するには、既存の値と新しい値のコンマ区切りのリストを指定します。

ユーザー権利プロファイルの管理

次の手順は、ローカルファイルを使用してシステム上のユーザー権利プロファイルを管理する方法を示しています。ネームサービスでユーザープロファイルを管理するには、『[System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)』を参照してください。

▼ 権利プロファイルをユーザーに割り当てる方法

LDoms Management プロファイルが直接割り当てられているユーザーは、セキュリティ属性を指定して `ldm` コマンドを実行するために、プロファイルシェルを起動する必要があります。詳細は、『[System Administration Guide: Security Services](#)』または『[Securing Users and Processes in Oracle Solaris 11.3](#)』を参照してください。

1. 管理者になります。

Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「[About Using Rights to Control Users and Processes](#)」を参照してください。

2. 管理プロファイルをローカルユーザーアカウントに割り当てます。

LDoms Review プロファイルまたは LDoms Management プロファイルのいずれかをユーザーアカウントに割り当てることができます。

```
# usermod -P "profile-name" username
```

次のコマンドは、LDoms Management プロファイルをユーザー `sam` に割り当てます。

```
# usermod -P "LDoms Management" sam
```

ユーザーへの役割の割り当て

次の手順は、ローカルファイルを使用して役割を作成し、ユーザーに割り当てる方法を示しています。ネームサービスで役割を管理するには、『[System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)』を参照してください。

この手順を使用する利点は、特定の役割が割り当てられたユーザーだけがその役割になれることです。役割にパスワードが割り当てられている場合は、その役割になるときにパスワードが必要です。次の 2 つのセキュリティ階層は、パスワードを保有するユーザーが、割り当てられていない役割になることを防止します。

▼ 役割を作成し、ユーザーにその役割を割り当てる方法

1. 管理者になります。

Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「About Using Rights to Control Users and Processes」を参照してください。

2. 役割を作成します。

```
# roleadd -P "profile-name" role-name
```

3. 役割にパスワードを割り当てます。

新しいパスワードを指定し、確認するようにプロンプトが表示されます。

```
# passwd role-name
```

4. ユーザーに役割を割り当てます。

```
# useradd -R role-name username
```

5. ユーザーにパスワードを割り当てます。

新しいパスワードを指定し、確認するようにプロンプトが表示されます。

```
# passwd username
```

6. 必要に応じてそのユーザーになり、パスワードを入力します。

```
# su username
```

7. ユーザーが割り当てられた役割にアクセスできることを確認します。

```
$ id
uid=nn(username) gid=nn(group-name)
$ roles
role-name
```

8. 必要に応じてその役割になり、パスワードを入力します。

```
$ su role-name
```

9. ユーザーがその役割になったことを確認します。

```
$ id
uid=nn(role-name) gid=nn(group-name)
```

例 2-1 役割の作成とユーザーへの割り当て

次の例では、ldm_read の役割を作成し、その役割を user_1 ユーザーに割り当てて user_1 ユーザーになり、ldm_read の役割を引き受けます。

```
# roleadd -P "LDoms Review" ldm_read
# passwd ldm_read
New Password:
Re-enter new Password:
passwd: password successfully changed for ldm_read
# useradd -R ldm_read user_1
# passwd user_1
New Password:
Re-enter new Password:
passwd: password successfully changed for user_1
# su user_1
Password:
$ id
uid=95555(user_1) gid=10(staff)
$ roles
ldm_read
$ su ldm_read
Password:
$ id
uid=99667(ldm_read) gid=14(sysadmin)
```

Logical Domains Manager プロファイルの内容

Logical Domains Manager パッケージは、次の権利プロファイルを、ローカルの権利プロファイル記述データベースに追加します。

```
LDoms Power Mgmt Observability::View LDoms Power Consumption:auths=solaris.ldoms.ldmpower
LDoms Review::Review LDoms configuration:profiles=LDoms Power Mgmt Observability;
auths=solaris.ldoms.read
LDoms Management::Manage LDoms domains:profiles=LDoms Power Mgmt Observability;auths=solaris.
ldoms.*
```

Logical Domains Manager パッケージは、LDoms Management プロファイルおよび LDoms Power Mgmt Observability プロファイルに関連付けられている次の実行属性も、ローカルの実行プロファイルデータベースに追加します。

```
LDoms Management:suser:cmd::/usr/sbin/ldm:privs=file_dac_read,file_dac_search
LDoms Power Mgmt Observability:suser:cmd::/usr/sbin/ldmpower:privs=file_dac_search
```

ldm サブコマンドと、そのコマンドの実行に必要な対応するユーザー承認を次の表に示します。

表 2-1 ldm サブコマンドおよびユーザー承認

ldm サブコマンド [†]	ユーザー承認
add-*	solaris.ldoms.write
bind-domain	solaris.ldoms.write
list	solaris.ldoms.read

ldm サブコマンド [†]	ユーザー承認
list-*	solaris.ldoms.read
panic-domain	solaris.ldoms.write
remove-*	solaris.ldoms.write
set-*	solaris.ldoms.write
start-domain	solaris.ldoms.write
stop-domain	solaris.ldoms.write
unbind-domain	solaris.ldoms.write

[†] 追加、表示、削除、または設定できるすべてのリソースを指します。

権利の使用によるドメインコンソールへのアクセスの制御

デフォルトでは、すべてのユーザーがすべてのドメインコンソールにアクセスできます。ドメインコンソールへのアクセスを制御するには、承認チェックを行うように `vntsd` デーモンを構成します。`vntsd` デーモンは、`vntsd/authorization` という名前のサービス管理機能 (SMF) プロパティを提供します。このプロパティを構成すると、ドメインコンソールまたはコンソールグループ用にユーザーおよび役割の承認チェックを有効にできます。承認チェックを有効にするには、`svccfg` コマンドを使用して、このプロパティの値を `true` に設定します。このオプションが有効な場合、`vntsd` は、`localhost` のみで接続を待機して受け入れます。`vntsd/authorization` が有効な場合、`listen_addr` プロパティに代替 IP アドレスを指定していても、`vntsd` は代替 IP アドレスを無視し、引き続き `localhost` のみで待機します。



注意 - `localhost` 以外のホストを使用するには、`vntsd` サービスを構成しないでください。

`localhost` 以外のホストを指定すると、制御ドメインからゲストドメインコンソールへの接続が制限されなくなります。ゲストドメインへのリモート接続に `telnet` コマンドを使用する場合は、平文のログイン認証がネットワーク上に渡されます。

デフォルトで、すべてのゲストコンソールへのアクセスの承認はローカルの承認記述データベースにあります。

```
solaris.vntsd.consoles::Access All LDoms Guest Consoles::
```

`usermod` コマンドを使用して、ローカルファイル上のユーザーまたは役割に必要な承認を割り当てます。このコマンドは、指定のドメインコンソールまたはコンソールグループへのアクセスに、

必要な承認を持つユーザーまたは役割のみを許可します。ネームサービスでユーザーまたは役割に承認を割り当てるには、『[System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)』を参照してください。

すべてのドメインコンソールまたは 1 つのドメインコンソールへのアクセスを制御できます。

- すべてのドメインコンソールへのアクセスを制御するには、[31 ページ](#)の「[役割を使用してすべてのドメインコンソールへのアクセスを制御する方法](#)」および[33 ページ](#)の「[権利プロファイルを使用してすべてのドメインコンソールへのアクセスを制御する方法](#)」を参照してください。
- 1 つのドメインコンソールへのアクセスを制御するには、[34 ページ](#)の「[役割を使用して 1 つのコンソールへのアクセスを制御する方法](#)」および[35 ページ](#)の「[権利プロファイルを使用して 1 つのコンソールへのアクセスを制御する方法](#)」を参照してください。

▼ 役割を使用してすべてのドメインコンソールへのアクセスを制御する方法

1. コンソールの承認チェックを有効にして、ドメインコンソールへのアクセスを制限します。

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. `solaris.vntsd.consoles` 承認を持つ役割を作成し、それによってすべてのドメインコンソールへのアクセスを許可します。

```
primary# roleadd -A solaris.vntsd.consoles role-name
primary# passwd role-name
```

3. ユーザーに新しい役割を割り当てます。

```
primary# usermod -R role-name username
```

例 2-2 役割の使用によるすべてのドメインコンソールへのアクセスの制御

最初に、コンソールの承認チェックを有効にして、すべてのドメインコンソールへのアクセスを制限します。

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
primary# ldm ls
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	8	16G	0.2%	47m
ldg1	active	-n--v-	5000	2	1G	0.1%	17h 50m
ldg2	active	-t----	5001	4	2G	25%	11s

次の例は、solaris.vntsd.consoles 承認を持つ all_cons 役割を作成し、それによってすべてのドメインコンソールへのアクセスを許可する方法を示しています。

```
primary# roleadd -A solaris.vntsd.consoles all_cons
primary# passwd all_cons
New Password:
Re-enter new Password:
passwd: password successfully changed for all_cons
```

このコマンドは、sam ユーザーに all_cons 役割を割り当てます。

```
primary# usermod -R all_cons sam
```

ユーザー sam は、all_cons 役割になり、すべてのコンソールにアクセスできます。たとえば、次のように表示されます。

```
$ id
uid=700299(sam) gid=1(other)
$ su all_cons
Password:
$ telnet localhost 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..

$ telnet localhost 5001
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg2" in group "ldg2" ....
Press ~? for control options ..
```

この例は、承認されていないユーザー dana がドメインコンソールへのアクセスを試みたときに何が起こるかを示しています。

```
$ id
uid=702048(dana) gid=1(other)
$ telnet localhost 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
Connection to 0 closed by foreign host.
```


▼ 権利プロファイルを使用してすべてのドメインコンソールへのアクセスを制御する方法

1. コンソールの承認チェックを有効にして、ドメインコンソールへのアクセスを制限します。

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. **solaris.vntsd.consoles** 承認を持つ権利プロファイルを作成します。

profiles コマンドを使用して、新しいプロファイルを作成します。

```
primary# profiles -p "LDoms Consoles" \
'set desc="Access LDoms Consoles"; set auths=solaris.vntsd.consoles'
```

3. 権利プロファイルをユーザーに割り当てます。

```
primary# usermod -P +"LDoms Consoles" username
```

4. ユーザーとしてドメインコンソールに接続します。

```
$ telnet localhost 5000
```

例 2-3 権利プロファイルの使用によるすべてのドメインコンソールへのアクセスの制御

次の例は、権利プロファイルを使用してすべてのドメインコンソールへのアクセスを制御する方法を示しています。profiles コマンドを使用して、solaris.vntsd.consoles 承認を持つ権利プロファイルを権利プロファイル記述データベース内に作成します。

```
primary# profiles -p "LDoms Consoles" \
'set desc="Access LDoms Consoles"; set auths=solaris.vntsd.consoles'
```

権利プロファイルをユーザーに割り当てます。

```
primary# usermod -P +"LDoms Consoles" sam
```

次のコマンドは、ユーザーが sam であり、All、Basic Solaris User、および LDoms Consoles 権利プロファイルが有効であることを確認する方法を示しています。telnet コマンドは、ldg1 ドメインコンソールにアクセスする方法を示します。

```
$ id
uid=702048(sam) gid=1(other)
$ profiles
All
Basic Solaris User
LDoms Consoles
$ telnet localhost 5000
```

```
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

▼ 役割を使用して 1 つのコンソールへのアクセスを制御する方法

1. コンソールの承認チェックを有効にして、ドメインコンソールへのアクセスを制限します。

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. 単一のドメインに対する承認を承認記述データベースに追加します。

承認名は、ドメインの名前から派生し、`solaris.vntsd.console-domain` の形式になります。

```
solaris.vntsd.console-domain::Access domain Console::
```

3. ドメインのコンソールへのアクセスのみを許可する、新しい承認を持つ役割を作成します。

```
primary# roleadd -A solaris.vntsd.console-domain role-name
primary# passwd role-name
New Password:
Re-enter new Password:
passwd: password successfully changed for role-name
```

4. `role-name` 役割をユーザーに割り当てます。

```
primary# usermod -R role-name username
```

例 2-4 1 つのドメインコンソールへのアクセス

この例は、ユーザー `terry` が `ldg1cons` 役割になり、`ldg1` ドメインコンソールにアクセスする方法を示しています。

まず、単一のドメイン `ldg1` に対する承認を承認記述データベースに追加します。

```
solaris.vntsd.console-ldg1::Access ldg1 Console::
```

次に、ドメインのコンソールへのアクセスのみを許可する、新しい承認を持つ役割を作成します。

```
primary# roleadd -A solaris.vntsd.console-ldg1 ldg1cons
primary# passwd ldg1cons
```

```
New Password:
Re-enter new Password:
passwd: password successfully changed for ldg1cons
```

ldg1cons 役割をユーザー terry に割り当てて、ldg1cons 役割になり、ドメインコンソールにアクセスします。

```
primary# usermod -R ldg1cons terry
primary# su terry
Password:
$ id
uid=700300(terry) gid=1(other)
$ su ldg1cons
Password:
$ id
uid=700303(ldg1cons) gid=1(other)
$ telnet localhost 5000
Trying 0.0.0.0...
Escape character is '^]'.
```

```
Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

次の例は、ユーザー terry が ldg2 ドメインコンソールにアクセスできないことを示しています。

```
$ telnet localhost 5001
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
Connection to 0 closed by foreign host.
```

▼ 権利プロファイルを使用して 1 つのコンソールへのアクセスを制御する方法

1. コンソールの承認チェックを有効にして、ドメインコンソールへのアクセスを制限します。

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. 単一のドメインに対する承認を承認記述データベースに追加します。

次の例のエントリは、ドメインコンソールに対する承認を追加します。

```
solaris.vntsd.console-domain:::Access domain Console:::
```

3. 特定のドメインコンソールにアクセスするための承認を持つ権利プロファイルを作成します。

profiles コマンドを使用して、新しいプロファイルを作成します。

```
primary# profiles -p "domain Console" \
'set desc="Access domain Console";
set auths=solaris.vntsd.console-domain'
```

4. 権利プロファイルを割り当てます。

```
primary# usermod -P +"domain Console" username
```

Logical Domains Manager の監査の有効化および使用

Logical Domains Manager は Oracle Solaris OS の監査機能を使用して、制御ドメインに発生したアクションおよびイベントの履歴を検査します。履歴は、何が、いつ、だれによって行われ、どのような影響があるかを追跡するログに保持されます。

注記 - Oracle VM Server for SPARC では、デフォルトでは Logical Domains Manager のアクションの監査レコードは生成されません。

Oracle Solaris OS の監査機能を有効および無効にできます。`audit` コマンドを使用します。`audit(1M)` のマニュアルページおよび『[Managing Auditing in Oracle Solaris 11.3](#)』を参照してください。

▼ Logical Domains Manager の監査を有効にする方法

Oracle Solaris 11 の監査はデフォルトで有効になっていますが、Logical Domains Manager の監査を構成する必要があります。

注記 - 既存のプロセスは、仮想化ソフトウェア (vs) クラスに対しては監査されません。この手順は、必ず通常のユーザーがシステムにログインする前に実行してください。

1. `/etc/security/audit_event` ファイルおよび `/etc/security/audit_class` ファイルにカスタマイズを追加します。

これらのカスタマイズは、Oracle Solaris のアップグレード全体に保持されますが、Oracle Solaris の新規インストール後は再度追加する必要があります。

a. 次のエントリが存在していない場合は、`audit_event` ファイルに追加します。

```
40700:AUE_ldoms:ldoms administration:vs
```

- b. 次のエントリが存在していない場合は、`audit_class` ファイルに追加します。

```
0x10000000:vs:virtualization_software
```

2. **vs 監査クラスを事前選択します。**

- a. **すでに選択されている監査クラスを確認します。**

すでに選択されているすべての監査クラスが、更新済みのクラスのセットの一部であることを確認します。次の例は、`lo` クラスがすでに選択されていることを示しています。

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
```

- b. **vs 監査クラスを追加します。**

```
# auditconfig -setflags [class],vs
```

`class` は、コンマで区切ったゼロ以上の監査クラスです。`/etc/security/audit_class` ファイルで、監査クラスのリストを確認できます。Oracle VM Server for SPARC システムに `vs` クラスを含めてください。

たとえば、次のコマンドは、`lo` および `vs` クラスの両方を選択します。

```
# auditconfig -setflags lo,vs
```

- c. (オプション) プロセスを、管理者または構成者のいずれかとして監査する場合は、システムからログアウトします。

ログアウトしない場合は、『[Managing Auditing in Oracle Solaris 11.3](#)』の「[How to Update the Preselection Mask of Logged In Users](#)」を参照してください。

3. **Oracle Solaris OS の監査が有効になっていることを確認します。**

```
# auditconfig -getcond
```

監査ソフトウェアが実行されている場合は、出力に `audit condition = auditing` が表示されます。

4. **監査レコードを生成するように Logical Domains Manager を構成します。**

- a. **ldmd/audit SMF プロパティ値を `true` に設定します。**

```
# svccfg -s ldmd setprop ldmd/audit = boolean: true
```

- b. `ldmd` サービスをリフレッシュします。

```
# svcadm refresh ldmd
```

- c. `ldmd` サービスを再起動します。

```
# svcadm restart ldmd
```

▼ Logical Domains Manager の監査を無効にする方法

1. `ldmd/audit` SMF プロパティ値を `false` に設定します。

```
# svccfg -s ldmd setprop ldmd/audit = boolean: false
```

2. `ldmd` サービスをリフレッシュします。

```
# svcadm refresh ldmd
```

3. `ldmd` サービスを再起動します。

```
# svcadm restart ldmd
```

▼ 監査レコードを確認する方法

- `vs` 監査の出力を確認するには、次のいずれかの方法を使用します。

- `auditreduce` コマンドおよび `praudit` コマンドを使用して、監査の出力を確認します。

```
# auditreduce -c vs | praudit
# auditreduce -c vs -a 20060502000000 | praudit
```

- `praudit -x` コマンドを使用して、監査レコードを XML 形式で出力します。

ドメインコンソールのロギングの使用

Oracle VM Server for SPARC 環境では、`primary` ドメインからのコンソール I/O はサービスプロセッサ (SP) に送信されます。ほかのすべてのドメインからのコンソール I/O は、仮想コ

ンソール端末集配信装置 (vcc) を実行するサービスドメインにリダイレクトされます。サービスドメインが Oracle Solaris 11 OS を実行する場合、ゲストドメインのコンソール出力をファイルに記録できます。

サービスドメインは、論理ドメインのコンソールロギングをサポートします。サービスドメインは Oracle Solaris 11 OS を実行する必要がありますが、ログの対象となっているゲストドメインは Oracle Solaris 10 OS と Oracle Solaris 11 OS のどちらを実行してもかまいません。

ドメインコンソールログは、vcc サービスを提供するサービスドメイン上の `/var/log/vntsd/domain/console-log` というファイルに保存されます。コンソールログファイルは `logadm` コマンドを使用して交替することができます。`logadm(1M)` および `logadm.conf(4)` のマニュアルページを参照してください。

Oracle VM Server for SPARC ソフトウェアでは、各論理ドメインのコンソールロギングを選択的に有効化および無効化できます。コンソールロギングはデフォルトで有効です。

▼ コンソールロギングを有効または無効にする方法

ドメインが同じコンソールグループに属している場合でも、個々の論理ドメインのコンソールロギングを有効化または無効化する必要があります。

1. ドメインの現在のコンソール設定をリストします。

```
primary# ldm list -o console domain
```

2. ドメインを停止し、バインドを解除します。

ドメインは、コンソール設定を変更する前に、アクティブではなくバインドされていない状態である必要があります。

```
primary# ldm stop domain
primary# ldm unbind domain
```

3. コンソールロギングを有効または無効にします。

■ コンソールロギングを有効にするには。

```
primary# ldm set-vcons log=on domain
```

■ コンソールロギングを無効にするには。

```
primary# ldm set-vcons log=off domain
```

ドメインコンソールのロギングに関するサービスドメインの要件

Oracle Solaris 11.1 より古い OS バージョンを実行するサービスドメインに接続されているドメインは、ログに記録することはできません。

注記 - ドメインのコンソールロギングを有効化した場合でも、ドメインの仮想コンソールは、必要なサポートがサービスドメインで使用可能でない場合は記録されません。

◆◆◆ 第 3 章

サービスおよび制御ドメインの設定

この章では、デフォルトのサービスおよび制御ドメインの設定に必要な手順について説明します。

この章では、次の項目について説明します。

- [41 ページの「出力メッセージ」](#)
- [42 ページの「デフォルトのサービスの作成」](#)
- [43 ページの「制御ドメインの初期構成」](#)
- [46 ページの「ドメインを使用するためのリブート」](#)
- [47 ページの「Oracle Solaris 10 サービスドメインとその他のドメインの間のネットワークの有効化」](#)
- [48 ページの「仮想ネットワーク端末サーバーデーモンの有効化」](#)
- [49 ページの「ILOM 相互接続が有効になっていることの確認」](#)

出力メッセージ

制御ドメインでリソースを動的に構成できない場合は、まず遅延再構成を開始することをお勧めします。遅延再構成は、制御ドメインのリブートが完了するまで構成処理を延期します。

primary ドメインで遅延再構成を開始すると、次のメッセージが表示されます。

```
Initiating a delayed reconfiguration operation on the primary domain.  
All configuration changes for other domains are disabled until the  
primary domain reboots, at which time the new configuration for the  
primary domain also takes effect.
```

primary ドメインをリブートするまで、その後の各操作のあとに次の通知を受け取ります。

```
Notice: The primary domain is in the process of a delayed reconfiguration.  
Any changes made to the primary domain will only take effect after it reboots.
```

デフォルトのサービスの作成

次の仮想デバイスサービスを作成し、制御ドメインをサービスドメインとして使用してほかのドメインの仮想デバイスを作成する必要があります。

- vcc – 仮想コンソール端末集配信装置サービス
- vds – 仮想ディスクサーバー
- vsw – 仮想スイッチサービス

▼ デフォルトのサービスを作成する方法

1. 仮想ネットワーク端末サーバーデーモン (vntsd) が使用する仮想コンソール端末集配信装置 (vcc) サービスを、すべての論理ドメインコンソールの端末集配信装置として作成します。

たとえば、次のコマンドを使用して、ポートの範囲が 5000 - 5100 の仮想コンソール端末集配信装置サービス (primary-vcc0) を、制御ドメイン (primary) に追加します。

```
primary# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

2. 論理ドメインに仮想ディスクをインポートできるように、仮想ディスクサーバー (vds) を作成します。

たとえば、次のコマンドを使用して、仮想ディスクサーバー (primary-vds0) を制御ドメイン (primary) に追加します。

```
primary# ldm add-vds primary-vds0 primary
```

3. 論理ドメインの仮想ネットワーク (vnet) デバイス間でネットワークを有効にするには、仮想スイッチサービス (vsw) を作成します。

各論理ドメインが仮想スイッチを使用して外部と通信する必要がある場合は、GLDv3 準拠のネットワークアダプタを仮想スイッチに割り当てます。

ゲストドメインのネットワークに使用するネットワークデバイス上に仮想スイッチサービス (primary-vsw0) を追加します。

```
primary# ldm add-vsw net-dev=network-device vsw-service primary
```

たとえば、次のコマンドを使用して、ネットワークデバイス net0 の仮想スイッチサービス (primary-vsw0) を、制御ドメイン (primary) に追加します。

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

ldm list-netdev -b コマンドを使用すると、仮想スイッチで利用できるバックエンドのネットワークデバイスを判定できます。[247 ページの「仮想スイッチ」](#)を参照してください。

net-dev プロパティの値は、ldm set-vsw コマンドを使用すると、直接更新できます。

4. list-services サブコマンドを使用して、サービスが作成されたことを確認します。

次のように出力されるはずです。

```
primary# ldm list-services primary
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0

VCC
  NAME          PORT-RANGE
  primary-vcc0  5000-5100

VSW
  NAME          MAC          NET-DEV          DEVICE          MODE
  primary-vsw0  02:04:4f:fb:9f:0d net0            switch@0        prog,promisc
```

制御ドメインの初期構成

最初に、すべてのシステムリソースが制御ドメインに割り当てられます。その他の論理ドメインを作成できるように、一部のリソースを解放する必要があります。

制御ドメインの構成

▼ 制御ドメインを構成する方法

この手順には、制御ドメイン用に設定するリソースの例も含まれています。ここで示す数値は単なる例であり、使用される値が制御ドメインに適していない場合があります。

ドメインのサイズ設定の推奨事項については、[Oracle VM Server for SPARC Best Practices \(http://www.oracle.com/technetwork/server-storage/vm/ovmsparc-best-practices-2334546.pdf\)](http://www.oracle.com/technetwork/server-storage/vm/ovmsparc-best-practices-2334546.pdf) を参照してください。

1. 仮想 CPU を制御ドメインに割り当てます。

制御ドメインを含むサービスドメインには、ゲストドメインに対する仮想ディスクや仮想ネットワークの I/O 操作を実行するための CPU およびメモリーリソースが必要です。割り当てる CPU およびメモリーリソースの量はゲストドメインのワークロードに応じて異なります。

たとえば、次のコマンドでは 2 つの CPU コア (16 の仮想 CPU スレッド) が制御ドメイン `primary` に割り当てられます。仮想 CPU スレッドの残りの部分はゲストドメインで使用できます。

```
primary# ldm set-core 2 primary
```

実際の CPU の割り当ては、アプリケーションの要件に基づいて動的に変更できます。制御ドメインの CPU 使用率を判定するには、`ldm list` コマンドを使用します。制御ドメインの CPU 使用率が高い場合は、`ldm add-core` および `ldm set-core` コマンドを使用して、サービスドメインに CPU リソースを追加します。

2. 制御ドメインに暗号化デバイスが必要かどうかを判断します。

暗号化デバイス (MAU) があるのは、UltraSPARC T2、UltraSPARC T2 Plus、および SPARC T3 プラットフォームのみです。SPARC T4 システムや Fujitsu M10 サーバーなどの新しいプラットフォームでは、すでに暗号化アクセラレーションが提供されているので、これらのプラットフォームに暗号化アクセラレータを割り当てる必要はありません。

以前のバージョンのプロセッサのいずれかを使用している場合、制御ドメイン内の各 CPU コア全体に対して 1 つの暗号化ユニットを割り当てます。

次の例では、2 つの暗号化リソースが制御ドメイン `primary` に割り当てられます。

```
primary# ldm set-crypto 2 primary
```

3. 制御ドメインの遅延再構成を開始します。

```
primary# ldm start-reconf primary
```

4. メモリーを制御ドメインに割り当てます。

たとえば、次のコマンドでは、16G バイトのメモリーが制御ドメイン `primary` に割り当てられます。この設定により、残りのメモリーをゲストドメインで使用できるようになります。

```
primary# ldm set-memory 16G primary
```

5. ドメイン構成をサービスプロセッサ (SP) に保存します。

たとえば、次のコマンドを使用して `initial` という名前の構成を追加します。

```
primary# ldm add-config initial
```

6. 次回のリブート時に構成が使用できる状態であることを確認します。

```
primary# ldm list-config
factory-default
initial [current]
```

この `ldm list-config` コマンドは、電源再投入の実行後に `initial` 構成セットが使用されることを示します。

7. 制御ドメインをリブートして、再構成の変更を有効にします。

CPU およびメモリーリソースを制御ドメインの `factory-default` 初期構成から縮小

CPU DR を使用すると、制御ドメインのコア数を `factory-default` 初期構成から減らすことができます。ただし、制御ドメインのメモリーを減らすには、メモリー DR ではなく遅延再構成を使用する必要があります。

`factory-default` 構成では、制御ドメインがホストシステムのメモリーをすべて所有します。アクティブなドメインは、要求されたメモリーのすべてを追加すること、またはより一般的には放棄することが保証されていないため、この目的にメモリー DR 機能は適していません。むしろ、そのドメインで実行されている OS が、要求を最大限に満たすことを試みます。さらに、メモリーの削除操作には長い時間が必要な場合があります。これらの問題は、制御ドメインのメモリーを最初に減らすときのように、大量のメモリー操作が関与する場合にはさらに悪化します。

注記 - Oracle Solaris OS が ZFS ファイルシステムにインストールされると、存在している物理メモリー量に基づいて、スワップ領域とダンプ領域が ZFS ルートプール内に ZFS ボリュームとして自動的にサイズ設定され、作成されます。ドメインのメモリー割り当てを変更した場合、これらのボリュームの推奨サイズが変更される可能性があります。制御ドメインのメモリーの縮小後に必要とされる量よりも、割り当て量が多い場合があります。スワップ領域の推奨事項については、『[Managing File Systems in Oracle Solaris 11.3](#)』の「[Planning for Swap Space](#)」を参照してください。ディスク領域を開放する前に、オプションでスワップおよびダンプ領域を変更できます。『[Managing ZFS File Systems in Oracle Solaris 11.3](#)』の「[Managing ZFS Swap and Dump Devices](#)」を参照してください。

▼ CPU およびメモリーリソースを制御ドメインの factory-default 初期構成から縮小する方法

この手順は、CPU およびメモリーリソースを制御ドメインの factory-default 初期構成から縮小する方法を示しています。まず CPU DR を使用してコア数を減らし、次に、メモリー量を減らす前に、遅延再構成を開始します。

例の値は、ldmd デーモンを実行して移行を実行するために十分な量のリソースがある、小さい制御ドメインの CPU およびメモリーサイズに対応しています。ただし、追加の目的で制御ドメインを使用する場合は、必要に応じて、より多くの数のコアやメモリー量を制御ドメインに割り当てることができます。

1. **factory-default 構成をブートします。**
2. **制御ドメインを構成します。**

[43 ページの「制御ドメインを構成する方法」](#)を参照してください。

ドメインを使用するためのリブート

構成の変更を有効にして、ほかの論理ドメインで使用できるようにリソースを解放するには、制御ドメインをリブートする必要があります。

▼ リブートする方法

- **制御ドメインを停止してリブートします。**

```
primary# shutdown -y -g0 -i6
```

注記 - リブートまたは電源再投入のいずれかによって、新しい構成がインスタンス化されます。サービスプロセッサ (SP) に保存されている構成が実際にブートされるのは、電源再投入後のみで、その際に list-config の出力に反映されます。

Oracle Solaris 10 サービスドメインとその他のドメインの間のネットワークの有効化

デフォルトでは、システム内の Oracle Solaris 10 サービスドメインとその他のドメインの間のネットワークは無効になっています。Oracle Solaris 10 OS ではネットワークがデフォルトでは有効になっていないため、仮想スイッチデバイスをネットワークデバイスとして構成することによってネットワークを有効にする必要があります。仮想スイッチは、基本となる物理デバイス (この例では `nxge0`) に代わりプライマリインタフェースとして構成するか、ドメインの追加のネットワークインタフェースとして構成することができます。

ゲストドメインは、対応するネットワークバックエンドデバイスが同じ仮想 LAN または仮想ネットワーク内に構成されているかぎり、Oracle Solaris 10 サービスドメインと自動的に通信できます。

▼ 仮想スイッチをプライマリインタフェースとして構成する方法

注記 - 次の手順は、ドメインへのネットワーク接続が一時的に中断される可能性があるため、Oracle Solaris 10 サービスドメインのコンソールから実行してください。

必要に応じて、物理ネットワークデバイスと同様に仮想スイッチを構成できます。この場合、手順 2 で記載されているように仮想スイッチを作成します。手順 3 はスキップして物理デバイスの削除をしません。そのあと、仮想スイッチは、静的 IP アドレスまたは動的 IP アドレスを使用して構成する必要があります。動的 IP アドレスは DHCP サーバーから取得できます。この場合の詳細および例は、[265 ページの「NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成」](#)を参照してください。

1. すべてのインタフェースのアドレス指定情報を出力します。

```
# ifconfig -a
```

2. 仮想スイッチネットワークインタフェースを構成します。

```
# ifconfig vsw0 plumb
```

3. 仮想スイッチ (`net-dev`) に割り当てられたデバイスの物理インタフェースを削除します。

```
# ifconfig nxge0 down unplumb
```

4. 物理ネットワークデバイス (nxge0) を仮想スイッチデバイス (vsw0) に移行するには、次のいずれかの操作を実行します。

- ネットワークが静的 IP アドレスを使用して構成されている場合は、仮想スイッチに対して nxge0 の IP アドレスとネットマスクを再利用します。

```
# ifconfig vsw0 IP-of-nxge0 netmask netmask-of-nxge0 broadcast + up
```

- ネットワークが DHCP を使用して構成されている場合は、仮想スイッチに対して DHCP を有効にします。

```
# ifconfig vsw0 dhcp start
```

5. 必要な構成ファイルに修正を加えて、この変更内容を確定します。

```
# mv /etc/hostname.nxge0 /etc/hostname.vsw0
# mv /etc/dhcp.nxge0 /etc/dhcp.vsw0
```

仮想ネットワーク端末サーバーデーモンの有効化

各論理ドメインの仮想コンソールにアクセスするには、仮想ネットワーク端末サーバーデーモン (vntsd) を有効にする必要があります。このデーモンの使用法の詳細は、vntsd(1M) マニュアルページを参照してください。

▼ 仮想ネットワーク端末サーバーデーモンを有効にする方法

注記 - vntsd を有効化する前に、必ず制御ドメイン上にデフォルトのサービス vconscon (vcc) を作成しておいてください。詳細は、[42 ページの「デフォルトのサービスの作成」](#)を参照してください。

1. 仮想ネットワーク端末サーバーデーモン vntsd を有効にします。

```
primary# svcadm enable vntsd
```

2. vntsd デーモンが有効であることを確認します。

```
primary# svcs vntsd
STATE      STIME      FMRI
online     Oct_08     svc:/ldoms/vntsd:default
```


ILOM 相互接続が有効になっていることの確認

ILOM 相互接続は `ldmd` デーモンと SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバー上のサービスプロセッサ (SP) の間の通信に必要であるため、無効にしてはいけません。詳細は、`ilomconfig(1M)` のマニュアルページを参照してください。

注記 - その他の SPARC T シリーズシステム、SPARC M5、および SPARC M6 システムでは、ILOM 相互接続を無効にすることは避けてください。ただし、そうした場合でも `ldmd` デーモンは SP と通信できます。

`ldm` コマンドを使用して SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバー上のドメイン構成を管理しようとして、SP との通信中のエラーのために失敗した場合は、ILOM 相互接続の状態をチェックし、必要に応じて `ilomconfig-interconnect` サービスを再度有効にします。[49 ページの「ILOM 相互接続構成を確認する方法」](#)および[50 ページの「ILOM 相互接続サービスを再度有効にする方法」](#)を参照してください。

▼ ILOM 相互接続構成を確認する方法

1. `ilomconfig-interconnect` サービスが有効になっていることを確認します。

```
primary# svcs ilomconfig-interconnect
STATE          STIME      FMRI
online          9:53:28    svc:/network/ilomconfig-interconnect:default
```

2. ILOM 相互接続が正しく構成されていることを確認します。

正しい構成では、「State」の値として `enabled` が表示され、「Host Interconnect IP Address」値として `none` ではなく、IP アドレスが表示されます。

```
primary# ilomconfig list interconnect
Interconnect
=====
State: enabled
Type: USB Ethernet
SP Interconnect IP Address: 169.254.182.76
Host Interconnect IP Address: 169.254.182.77
Interconnect Netmask: 255.255.255.0
SP Interconnect MAC Address: 02:21:28:57:47:16
Host Interconnect MAC Address: 02:21:28:57:47:17
```

3. `ldmd` デーモンが SP と通信できることを確認します。

```
primary# ldm list-spconfig
```

▼ ILOM 相互接続サービスを再度有効にする方法

`ilomconfig-interconnect` サービスはデフォルトで有効です。このサービスを手動で再度有効にする必要がある場合は、この手順を使用します。

1. ILOM 相互接続サービスを有効にします。

```
primary# svcadm enable ilomconfig-interconnect
```

2. `ilomconfig-interconnect` サービスが有効になっていることを確認します。

```
primary# svcs ilomconfig-interconnect
STATE      STIME      FMRI
online      9:53:28    svc:/network/ilomconfig-interconnect:default
```

3. ILOM 相互接続が正しく構成されていることを確認します。

正しい構成では、「State」の値として `enabled` が表示され、「Host Interconnect IP Address」値として `none` ではなく、IP アドレスが表示されます。

```
primary# ilomconfig list interconnect
Interconnect
=====
State: enabled
Type: USB Ethernet
SP Interconnect IP Address: 169.254.182.76
Host Interconnect IP Address: 169.254.182.77
Interconnect Netmask: 255.255.255.0
SP Interconnect MAC Address: 02:21:28:57:47:16
Host Interconnect MAC Address: 02:21:28:57:47:17
```

4. `ldmd` デーモンが SP と通信できることを確認します。

```
primary# ldm list-spcnfig
```

ゲストドメインの設定

この章では、ゲストドメインの設定に必要な手順について説明します。

この章では、次の項目について説明します。

- [51 ページの「ゲストドメインの作成と起動」](#)
- [55 ページの「ゲストドメインへの Oracle Solaris OS のインストール」](#)

ゲストドメインの作成と起動

ゲストドメインでは、sun4v プラットフォームとハイパーバイザによって提供される仮想デバイスの両方と互換性のあるオペレーティングシステムを実行する必要があります。現時点では、Oracle Solaris 10 11/06 OS 以降を実行する必要があります。Oracle Solaris 10 1/13 OS を実行すると、すべての Oracle VM Server for SPARC 3.3 機能を使用できます。必要になる可能性がある特定のパッチについては、『[Oracle VM Server for SPARC 3.3 インストールガイド](#)』を参照してください。デフォルトのサービスを作成し、制御ドメインからリソースを再度割り当てたら、ゲストドメインを作成して起動できます。

注記 - CPU の割り当て数が 1024 を超えているゲストドメインでは、Oracle Solaris 10 OS を実行できません。また、Oracle Solaris 10 OS を実行するために、CPU DR を使用して CPU の数を 1024 以下に減らすこともできません。

この問題を回避するには、そのゲストドメインをバインド解除し、CPU の数が 1024 以下になるまで CPU を削除してから、ゲストドメインを再バインドします。その後、このゲストドメインで Oracle Solaris 10 OS を実行できます。

▼ ゲストドメインを作成および起動する方法

1. 論理ドメインを作成します。

次のコマンドを使用して `ldg1` という名前のゲストドメインを作成します。

```
primary# ldm add-domain ldg1
```

2. CPU をゲストドメインに追加します。

次のいずれかの手順を実行します。

■ 仮想 CPU を追加します。

次のコマンドを使用して 8 つの仮想 CPU をゲストドメイン `ldg1` に追加します。

```
primary# ldm add-vcpu 8 ldg1
```

■ コア全体を追加します。

次のコマンドを使用して 2 つのコア全体をゲストドメイン `ldg1` に追加します。

```
primary# ldm add-core 2 ldg1
```

3. メモリーをゲストドメインに追加します。

次のコマンドを使用して 2G バイトのメモリーをゲストドメイン `ldg1` に追加します。

```
primary# ldm add-memory 2G ldg1
```

4. 仮想ネットワークデバイスをゲストドメインに追加します。

次のコマンドを使用して、次のように指定した仮想ネットワークデバイスをゲストドメイン `ldg1` に追加します。

```
primary# ldm add-vnet vnet1 primary-vsw0 ldg1
```

ここでは:

- `vnet1` は、後続の `set-vnet` または `remove-vnet` サブコマンドで参照するためにこの仮想ネットワークデバイスのインスタンスに割り当てられる、論理ドメインで一意的なインターフェース名です。

- `primary-vsw0` は、接続する既存のネットワークサービス (仮想スイッチ) の名前です。

注記 - 手順 5 および 6 は、仮想ディスクサーバーデバイス (`vdsdev`) を primary ドメインに、および仮想ディスク (`vdisk`) をゲストドメインに追加するための簡略化された方法です。ZFS ボリュームおよびファイルシステムを仮想ディスクとして使用する方法については、[192 ページの「ZFS ボリュームを 1 つのスライスディスクとしてエクスポートする方法」](#)および[205 ページの「仮想ディスクと ZFS の使用」](#)を参照してください。

5. 仮想ディスクサーバーによってゲストドメインに仮想ディスクとしてエクスポートされるデバイスを指定します。

物理ディスク、ディスクスライス、ボリューム、またはファイルをブロック型デバイスとしてエクスポートできます。物理ディスクとファイルの例を次に示します。

- **物理ディスクの例。**この例では、次の指定で物理ディスクを追加します。

```
primary# ldm add-vdsdev /dev/dsk/c2t1d0s2 vol1@primary-vds0
```

ここでは:

- `/dev/dsk/c2t1d0s2` は、実際の物理デバイスのパス名です。デバイスを追加する場合、パス名にはデバイス名を組み合わせる必要があります。
- `vol1` は、仮想ディスクサーバーに追加するデバイスに指定する必要がある一意の名前です。ボリューム名は、この仮想ディスクサーバーによってクライアントにエクスポートされ追加されるため、ボリューム名はこの仮想ディスクサーバーのインスタンスに対して一意である必要があります。デバイスを追加する場合、ボリューム名には実際のデバイスのパス名を組み合わせる必要があります。
- `primary-vds0` は、このデバイスを追加する仮想ディスクサーバーの名前です。
- **ファイルの例。**この例では、ファイルをブロック型デバイスとしてエクスポートします。

```
primary# ldm add-vdsdev backend vol1@primary-vds0
```

ここでは:

- `backend` は、ブロック型デバイスとしてエクスポートされる実際のファイルのパス名です。デバイスを追加する場合、このバックエンドにデバイス名を組み合わせる必要があります。
- `vol1` は、仮想ディスクサーバーに追加するデバイスに指定する必要がある一意の名前です。ボリューム名は、この仮想ディスクサーバーによってクライアントにエクスポートされ追加されるため、ボリューム名はこの仮想ディスクサーバーのインスタンスに対して一意である必要があります。デバイスを追加する場合、ボリューム名には実際のデバイスのパス名を組み合わせる必要があります。
- `primary-vds0` は、このデバイスを追加する仮想ディスクサーバーの名前です。

6. 仮想ディスクをゲストドメインに追加します。

次の例では、仮想ディスクをゲストドメイン `ldg1` に追加します。

```
primary# ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1
```

ここでは:

- `vdisk1` は、仮想ディスクの名前です。
- `vol1` は、接続する既存のボリュームの名前です。
- `primary-vds0` は、接続する既存の仮想ディスクサーバーの名前です。

注記 - 仮想ディスクは、さまざまな種類の物理デバイス、ボリューム、またはファイルに関連付けられた総称的なブロック型デバイスです。仮想ディスクは SCSI ディスクと同義ではありません。そのため、ディスクラベル内のターゲット ID は除外されます。論理ドメインの仮想ディスクの形式は、`cNdNsN` です。`cN` は仮想コントローラ、`dN` は仮想ディスク番号、および `sN` はスライスを示します。

7. **ゲストドメインの `auto-boot?` および `boot-device` 変数を設定します。**

次のコマンド例では、ゲストドメイン `ldg1` の `auto-boot?` を `true` に設定します。

```
primary# ldm set-var auto-boot\?=true ldg1
```

次のコマンド例では、ゲストドメイン `ldg1` の `boot-device` を `vdisk1` に設定します。

```
primary# ldm set-var boot-device=vdisk1 ldg1
```

8. **ゲストドメイン `ldg1` にリソースをバインドし、ドメインを一覧表示してリソースがバインドされていることを確認します。**

```
primary# ldm bind-domain ldg1
primary# ldm list-domain ldg1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	bound	-----	5000	8	2G		

9. **ゲストドメインのコンソールのポートを見つけるために、前述の `list-domain` サブコマンドの出力を調べます。**

`CONS` という見出しの下で、論理ドメインゲスト 1 (`ldg1`) のコンソール出力がポート `5000` にバインドされていることがわかります。

10. **制御ドメインにログインし、ローカルホストのコンソールポートに直接接続することによって、別の端末からゲストドメインのコンソールに接続します。**

```
$ ssh hostname.domain-name
$ telnet localhost 5000
```

11. **ゲストドメイン `ldg1` を起動します。**

```
primary# ldm start-domain ldg1
```

ゲストドメインへの Oracle Solaris OS のインストール

このセクションでは、ゲストドメインに Oracle Solaris OS をインストールできる、いくつかの異なる方法について説明します。



注意 - Oracle Solaris OS のインストール中に仮想コンソールの接続を解除しないでください。

Oracle Solaris 11 ドメインの場合は、DefaultFixed ネットワーク構成プロファイル (NCP) を使用します。このプロファイルは、インストール中またはインストール後に有効にできます。

Oracle Solaris 11 のインストール中に、「手動」ネットワーク構成を選択します。Oracle Solaris 11 のインストール後に、`netadm list` コマンドを使用して、DefaultFixed NCP が有効になっていることを確認してください。『[Configuring and Managing Network Components in Oracle Solaris 11.3](#)』の第 2、3、5、および 6 章を参照してください。

メモリーサイズの要件

Oracle VM Server for SPARC ソフトウェアでは、ドメインを作成するときのメモリーサイズの制限はありません。メモリーサイズの要件は、ゲストオペレーティングシステム特有のもので、Oracle VM Server for SPARC の機能によっては、現在のメモリー容量が推奨サイズより少ないと動作しない場合があります。Oracle Solaris 10 OS の推奨されるメモリー要件と最小メモリー要件については、『[Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade](#)』の「[System Requirements and Recommendations](#)」を参照してください。Oracle Solaris 11 OS の推奨されるメモリー要件と最小メモリー要件については、『[Oracle Solaris 11 Release Notes](#)』、『[Oracle Solaris 11.1 Release Notes](#)』、『[Oracle Solaris 11.2 Release Notes](#)』、および『[Oracle Solaris 11.3 Release Notes](#)』を参照してください。

OpenBoot PROM には、ドメインの最小サイズの制限があります。現在、制限値は 12M バイトです。このサイズより小さいドメインが存在すると、Logical Domains Manager はそのドメインのサイズを自動的に 12M バイトに引き上げます。Fujitsu M10 サーバーの最小サイズ制限は 256M バイトです。メモリーサイズの要件については、使用しているシステムファームウェアのリリースノートを参照してください。

メモリー動的再構成 (Dynamic Reconfiguration、DR) 機能では、操作に關与するメモリーのアドレスとサイズが 256M バイト単位であることが要求されます。[359 ページの「メモリー配置」](#)を参照してください。

▼ DVD からゲストドメインに Oracle Solaris OS をインストールする方法

1. DVD ドライブに Oracle Solaris 10 OS または Oracle Solaris 11 OS DVD を挿入します。
2. **primary** ドメインでボリューム管理デーモン **vold(1M)** を停止します。

```
primary# svcadm disable volfs
```

3. ゲストドメイン (**ldg1**) を停止し、バインドを解除します。

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

4. セカンダリボリュームおよび仮想ディスクとして、DVD-ROM メディアに DVD を追加します。

次の例では、**c0t0d0s2** を Oracle Solaris メディアを格納している DVD ドライブ、**dvd_vol@primary-vds0** をセカンダリボリューム、**vdisk_cd_media** を仮想ディスクとして使

```
primary# ldm add-vdsdev options=ro /dev/dsk/c0t0d0s2 dvd_vol@primary-vds0
primary# ldm add-vdisk vdisk_cd_media dvd_vol@primary-vds0 ldg1
```

5. DVD がセカンダリボリュームおよび仮想ディスクとして追加されていることを確認します。

```
primary# ldm list-bindings
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv	SP	8	8G	0.2%	22h 45m
...							
VDS							
NAME	VOLUME	OPTIONS		DEVICE			
primary-vds0	vol1			/dev/dsk/c2t1d0s2			
dvd_vol				/dev/dsk/c0t0d0s2			
....							

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	inactive	-----		60	6G		
...							
DISK							
NAME	VOLUME			TOUT	DEVICE	SERVER	
vdisk1	vol1@primary-vds0						


```
vdisk_cd_media    dvd_vol@primary-vds0
....
```

6. ゲストドメイン (ldg1) をバインドし、起動します。

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

7. クライアント OpenBoot PROM でデバイス別名を表示します。

この例で、Oracle Solaris DVD の vdisk_cd_media および Oracle Solaris OS をインストール可能な仮想ディスクの vdisk1 のデバイス別名を確認してください。

```
ok devalias
vdisk_cd_media  /virtual-devices@100/channel-devices@200/disk@1
vdisk1         /virtual-devices@100/channel-devices@200/disk@0
vnet1          /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name           aliases
```

8. ゲストドメインのコンソールで、スライス f の vdisk_cd_media (disk@1) からブートします。

```
ok boot vdisk_cd_media:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f  File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c), 1983-2010, Oracle and/or its affiliates. All rights reserved.
```

9. 引き続き Oracle Solaris OS のインストールに従います。

▼ Oracle Solaris ISO ファイルからゲストドメインに Oracle Solaris OS をインストールする方法

1. ゲストドメイン (ldg1) を停止し、バインドを解除します。

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

2. セカンダリボリュームおよび仮想ディスクとして、Oracle Solaris ISO ファイルを追加します。

次の例では、solarisdvd.iso を Oracle Solaris ISO ファイル、iso_vol@primary-vds0 をセカンダリボリューム、vdisk_iso を仮想ディスクとして使用します。

```
primary# ldm add-vdsdev /export/solarisdvd.iso iso_vol@primary-vds0
primary# ldm add-vdisk vdisk_iso iso_vol@primary-vds0 ldg1
```

3. Oracle Solaris ISO ファイルがセカンダリボリュームおよび仮想ディスクとして追加されていることを確認します。

```
primary# ldm list-bindings
NAME                STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary             active -n-cv  SP    8     8G      0.2%  22h 45m
...
VDS
  NAME              VOLUME              OPTIONS              DEVICE
  primary-vds0      voll                /dev/dsk/c2t1d0s2
  iso_vol            /export/solarisdvd.iso
....
-----
NAME                STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1                 inactive -----  60    6G
...
DISK
  NAME              VOLUME              TOUT ID DEVICE  SERVER  MPGROUP
  vdisk1            voll@primary-vds0
  vdisk_iso         iso_vol@primary-vds0
....
```

4. ゲストドメイン (ldg1) をバインドし、起動します。

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^J'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

5. クライアント OpenBoot PROM でデバイス別名を表示します。

この例で、vdisk_iso (Oracle Solaris ISO イメージ) および vdisk_install (ディスク領域) のデバイス別名を確認してください。

```
ok devalias
vdisk_iso          /virtual-devices@100/channel-devices@200/disk@1
vdisk1             /virtual-devices@100/channel-devices@200/disk@0
vnet1              /virtual-devices@100/channel-devices@200/network@0
virtual-console    /virtual-devices/console@1
```

name	aliases

6. ゲストドメインのコンソールで、スライス f の `vdisk_iso (disk@1)` からブートします。

```
ok boot vdisk_iso:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.
```

7. 引き続き Oracle Solaris OS のインストールに従います。

▼ Oracle Solaris 10 ゲストドメインで Oracle Solaris JumpStart 機能を使用する方法

注記 - Oracle Solaris JumpStart 機能は Oracle Solaris 10 OS でのみ利用可能です。[『Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations』](#)を参照してください。

Oracle Solaris 11 OS の自動インストールを実行するには、Automated Installer (AI) 機能を使用できます。[『Transitioning From Oracle Solaris 10 to Oracle Solaris 11.3』](#)を参照してください。

- **JumpStart プロファイルを変更して、ゲストドメインの異なるディスクデバイス名の形式を反映するようにします。**

論理ドメインの仮想ディスクデバイス名は、物理ディスクデバイス名とは異なります。仮想ディスクデバイス名には、ターゲット ID (tN) は含まれません。通常の `cNtNdNsN` 形式の代わりに、仮想ディスクデバイス名は `cNdNsN` という形式になります。cN は仮想コントローラ、dN は仮想ディスク番号、sN はスライス番号を示します。

注記 - フルディスクまたは 1 つのスライスディスクとして、仮想ディスクを表示できます。複数のパーティションを指定する通常の JumpStart プロファイルを使用して、フルディスクに Oracle Solaris OS をインストールできます。1 つのスライスディスクには、ディスク全体を使用する `s0` という 1 つのパーティションのみがあります。1 つのディスクに Oracle Solaris OS をインストールするには、ディスク全体を使用する 1 つのパーティション (/) を含むプロファイルを使用する必要があります。スワップなどのほかのパーティションは定義できません。フルディスクと 1 つのスライスディスクの詳細については、[183 ページの「仮想ディスクの表示」](#)を参照してください。

■ UFS ルートファイルシステムをインストールする JumpStart プロファイル

『[Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#)』を参照してください。

通常の UFS プロファイル

```
filesys c1t1d0s0 free /  
filesys c1t1d0s1 2048 swap  
filesys c1t1d0s5 120 /spare1  
filesys c1t1d0s6 120 /spare2
```

フルディスクにドメインをインストールする実際の UFS プロファイル

```
filesys c0d0s0 free /  
filesys c0d0s1 2048 swap  
filesys c0d0s5 120 /spare1  
filesys c0d0s6 120 /spare2
```

1 つのスライスディスクにドメインをインストールする実際の UFS プロファイル

```
filesys c0d0s0 free /
```

■ ZFS ルートファイルシステムをインストールする JumpStart プロファイル

『[Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#)』の第 9 章「[Installing a ZFS Root Pool With JumpStart](#)」を参照してください。

通常の ZFS プロファイル

```
pool rpool auto 2G 2G c1t1d0s0
```

ドメインをインストールする実際の ZFS プロファイル

```
pool rpool auto 2G 2G c0d0s0
```

I/O ドメインの構成

この章では、I/O ドメインについて、および Oracle VM Server for SPARC 環境で I/O ドメインを構成する方法について説明します。

この章では、次の項目について説明します。

- [61 ページの「I/O ドメインの概要」](#)
- [62 ページの「I/O ドメインを作成するための一般的なガイドライン」](#)

I/O ドメインの概要

I/O ドメインは、物理 I/O デバイスを直接所有し、物理 I/O デバイスに直接アクセスできます。作成するには、PCI EXPRESS (PCIe) バス、PCIe エンドポイントデバイス、または PCIe SR-IOV 仮想機能をドメインに割り当てます。ldm add-io コマンドを使用して、バス、デバイス、または仮想機能をドメインに割り当てます。

次のような理由で、I/O ドメインの構成が必要になることがあります。

- I/O ドメインは物理 I/O デバイスに直接アクセスできるため、仮想 I/O に関連するパフォーマンスオーバーヘッドを回避できます。その結果、I/O ドメインの I/O パフォーマンスは、基本的なシステムの I/O パフォーマンスにより近いものになります。
- I/O ドメインに仮想 I/O サービスをホストし、ゲストドメインがそのサービスを使用できるようにすることができます。

I/O ドメインの構成の詳細は、次の章の情報を参照してください。

- [第6章「PCIe バスの割り当てによってルートドメインを作成する方法」](#)
- [第8章「直接 I/O を使用した I/O ドメインの作成」](#)
- [第7章「PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成」](#)
- [第9章「primary 以外のルートドメインの使用」](#)

注記 - PCIe バス、PCIe エンドポイントデバイス、または SR-IOV 仮想機能を含むドメインは移行できません。移行に関するその他の制限については、[第13章「ドメインの移行」](#)を参照してください。

I/O ドメインを作成するための一般的なガイドライン

I/O ドメインは、PCIe バス、ネットワークインタフェースユニット (NIU)、PCIe エンドポイントデバイス、および PCIe シングルルート I/O 仮想化 (SR-IOV) の仮想機能など、1 つ以上の I/O デバイスに直接アクセスできる場合があります。

この種類の I/O デバイスへの直接アクセスは、次のことを提供するためにより多くの I/O 帯域幅を使用できることを意味します。

- I/O ドメイン内のアプリケーションへのサービスの提供
- ゲストドメインへの仮想 I/O サービスの提供

次の基本的なガイドラインにより、I/O 帯域幅を効率的に使用することができます。

- CPU リソースを CPU コアの細分度で割り当てます。I/O デバイスの種類と I/O ドメイン内の I/O デバイスの数に基づいて、1 つ以上の CPU コアを割り当てます。
たとえば、1G ビット/秒の Ethernet デバイスは、10G ビット/秒の Ethernet デバイスと比べて、より少ない CPU コアで全帯域幅を使用できる可能性があります。
- メモリー要件を順守します。メモリー要件は、ドメインに割り当てる I/O デバイスの種類によって異なります。I/O デバイスごとに最低 4 ギガバイト以上が推奨されます。割り当てる I/O デバイスの数が多いほど、割り当てる必要があるメモリーが多くなります。
- PCIe SR-IOV 機能を使用するときは、ほかの I/O デバイスで使用する可能性のある各 SR-IOV 仮想機能に対するガイドラインと同じガイドラインに従います。そのため、仮想機能から使用できる帯域幅を最大限に使用するには、1 つ以上の CPU コアとギガビット単位のメモリーを割り当てます。

多数の仮想機能を作成して、十分な CPU およびメモリーリソースがないドメインに割り当てると、最適な構成にならない可能性があることに注意してください。

SPARC T5 および SPARC M6 プラットフォームまでの SPARC システムでは、提供される割り込み数が制限されているので、Oracle Solaris は各デバイスが使用可能な割り込み数を制限します。デフォルトの制限は一般的なシステム構成のニーズに一致する必要がありますが、特定のシステム構成ではこの値の調整が必要になることがあります。[421 ページの「割り込み制限の調整」](#)を参照してください。

PCIe バスの割り当てによってルートドメインを作成する方法

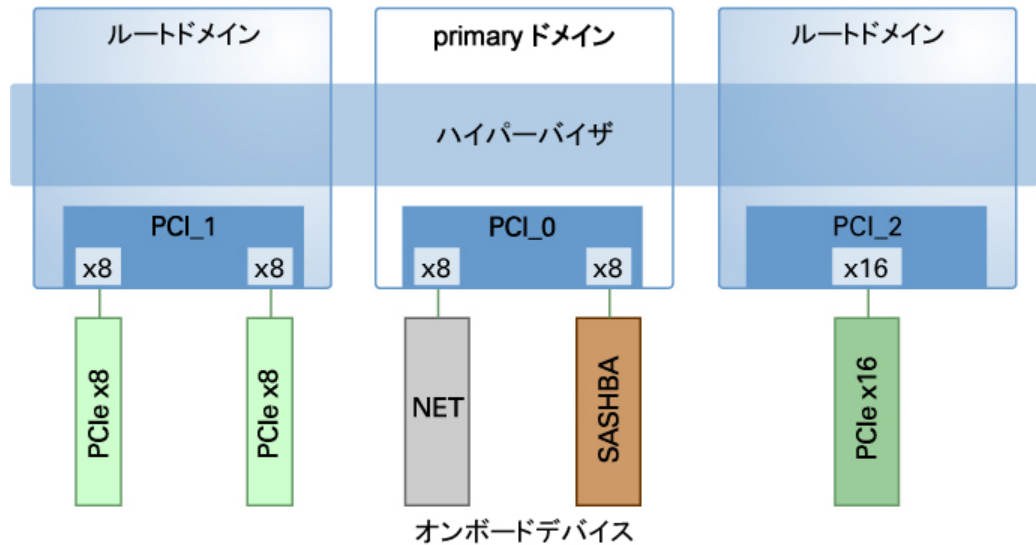
この章では、PCIe バスを割り当てることによって、ルートドメインを作成する方法について説明します。

PCIe バスの割り当てによってルートドメインを作成する方法

Oracle VM Server for SPARC ソフトウェアを使用して、PCIe バス全体 (別名「ルートコンプレックス」) をドメインに割り当てることができます。PCIe バス全体は、PCIe バス自体と、そのすべての PCI スイッチとデバイスで構成されます。サーバーに存在する PCIe バスは、`pci@400` (`pci_0`) などの名前により識別されます。PCIe バス全体で構成された I/O ドメインは、「ルートドメイン」とも呼ばれます。

次の図は、3 つのルートコンプレックス (`pci_0`, `pci_1`, `pci_2`) が存在するシステムを示しています。

図 6-1 PCIe バスのルートドメインへの割り当て



PCIe バスで作成できるルートドメインの最大数は、サーバー上で使用できる PCIe バスの数に依存します。システムで利用可能な PCIe バスの数を確認するには、`ldm list-io` を使用します。

PCIe バスをルートドメインに割り当てると、そのバス上のすべてのデバイスはそのルートに所有されます。そのバス上の任意の PCIe エンドポイントデバイスをほかのドメインに割り当てることができます。

Oracle VM Server for SPARC 環境でサーバーが最初に構成されるとき、または `factory-default` 構成を使用しているときに、`primary` ドメインはすべての物理デバイスリソースにアクセスできます。したがって、システムに構成されているルートドメインは `primary` ドメインのみであり、このドメインがすべての PCIe バスを所有します。

静的 PCIe バスの割り当て

ルートドメインでの静的 PCIe バスの割り当て方法では、PCIe バスの割り当てまたは削除を行うときに、まず、ルートドメインの遅延再構成を開始する必要があります。PCIe バスを所有し

ていないドメインにこの方法を使用する場合には、PCIe バスを割り当てる前にドメインを停止する必要があります。ルートドメインの構成手順を完了したら、それをリブートする必要があります。Oracle VM Server for SPARC 3.2 ファームウェアがシステムにインストールされていないか、各ドメインにインストールされている OS のバージョンが 動的 PCIe バスの割り当て をサポートしていない場合、この静的な方法を使用する必要があります。

ルートドメインの停止中または遅延再構成の実行時には、ルートドメインをリブートする前に `ldm add-io` および `ldm remove-io` コマンドを 1 つ以上実行できます。ドメインのダウンタイムを最小限に抑えるために、PCIe バスの割り当てまたは削除の前には計画を立ててください。

- ルートドメインでは、`primary` と `primary` 以外の両方で、遅延再構成を使用します。PCIe バスを追加または削除したら、変更を有効にするためにルートドメインをリブートします。

```
primary# ldm start-reconf root-domain
Add or remove the PCIe bus by using the ldm add-io or ldm remove-io command
primary# ldm stop -r domain-name
```

遅延再構成を使用できるのは、ドメインがすでに PCIe バスを所有している場合のみです。

- ルートドメイン以外では、ドメインを停止してから、PCIe バスを追加または削除します。

```
primary# ldm stop domain-name
Add or remove the PCIe bus by using the ldm add-io or ldm remove-io command
primary# ldm start domain-name
```

動的 PCIe バスの割り当て

動的 PCIe バスの割り当て 機能では、ルートドメインから PCIe バスを動的に割り当てたり、削除したりできます。

動的 PCIe バスの割り当て 機能は、システムで必要なファームウェアとソフトウェアが実行されている場合に有効になります。[66 ページの「動的 PCIe バスの割り当て の要件」](#)を参照してください。システムで必要なファームウェアとソフトウェアが実行されていない場合、`ldm add-io` および `ldm remove-io` コマンドはすみやかに失敗します。

有効になっている場合は、ルートドメインを停止したり遅延再構成したりしなくても、`ldm add-io` および `ldm remove-io` コマンドを実行できます。

動的 PCIe バスの割り当て の要件

動的 PCIe バスの割り当て機能は、ルートドメインで Oracle Solaris 11 OS を実行している SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、SPARC T7 シリーズサーバー、および Fujitsu M10 サーバーでサポートされています。SPARC M5 サーバーおよび SPARC M6 サーバーは少なくとも 9.4.2 バージョンのシステムファームウェアを実行している必要があり、SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーは少なくとも 9.4.3 を実行している必要があり、Fujitsu M10 サーバーは少なくとも XCP2240 を実行している必要があります。

▼ PCIe バスの割り当てによってルートドメインを作成する方法

この手順例は、複数のバスが primary ドメインによって所有されている場合に、初期構成から新しいルートドメインを作成する方法を示しています。デフォルトでは、システム上に存在するすべてのバスを primary ドメインが所有しています。この例は、SPARC T4-2 サーバーを対象としています。この手順は、ほかのサーバーにも使用できます。別のサーバーではこれらの手順と若干異なる場合がありますが、この例では基本的な方針について理解できます。

ブートディスクおよびプライマリネットワークインタフェースをホストする PCIe バスは primary ドメインから削除しないでください。



注意 - サポートされているサーバーの内部ディスクはすべて、単一の PCIe バスに接続されている場合があります。ドメインが内部ディスクからブートする場合は、ドメインからそのバスを削除しないでください。

ドメインによって使用されているデバイス (ネットワークポートや `usbcm` デバイスなど) を含むバスを削除しないようにしてください。誤ったバスを削除すると、ドメインは必要なデバイスにアクセスできず、使用できなくなることがあります。ドメインで使用されているデバイスが接続されたバスを削除する場合は、ほかのバスのデバイスを使用するよう、そのドメインを再構成してください。たとえば、別のオンボードネットワークポートや、別の PCIe スロットの PCIe カードを使用するよう、ドメインを再構成する必要がある場合があります。

特定の SPARC サーバーでは、USB、グラフィックコントローラ、およびその他のデバイスを含む PCIe バスを削除できます。ただし、これらの PCIe バスをほかのドメインに追加することはできません。このような PCIe バスは、primary ドメインにのみ追加できます。

この例では、primary ドメインは 1 つの ZFS プール (rpool) と 1 つのネットワークインタフェース (igb0) のみを使用します。primary ドメインで複数のデバイスを使用する場合は、デバイスごとに手順 2 - 4 を繰り返して、削除するバスにそれらのデバイスがないことを確認します。

ドメインにバスを追加したり、ドメインからバスを削除したりするには、そのデバイスパス (pci@nnn) またはその仮名 (pci_n) を使用します。ldm list-bindings primary または ldm list -l -o physio primary コマンドは次を示しています。

- pci@400 は pci_0 に対応します
- pci@500 は pci_1 に対応します
- pci@600 は pci_2 に対応します
- pci@700 は pci_3 に対応します

1. primary ドメインが複数の PCIe バスを所有していることを確認します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	
pci_1	BUS	pci_1	primary	
pci_2	BUS	pci_2	primary	
pci_3	BUS	pci_3	primary	
/SYS/MB/PCIE1	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA0	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE5	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE6	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE7	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE2	PCIE	pci_2	primary	EMP
/SYS/MB/PCIE3	PCIE	pci_2	primary	EMP
/SYS/MB/PCIE4	PCIE	pci_2	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_3	primary	EMP
/SYS/MB/SASHBA1	PCIE	pci_3	primary	OCC
/SYS/MB/NET2	PCIE	pci_3	primary	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_3	primary	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_3	primary	

2. 保持する必要があるブートディスクのデバイスパスを確認します。

- UFS ファイルシステムの場合、df / コマンドを実行して、ブートディスクのデバイスパスを確認します。

```
primary# df /
```

```
/ (/dev/dsk/c0t5000CCA03C138904d0s0):22755742 blocks 2225374 files
```

- ZFS ファイルシステムの場合、まず `df /` コマンドを実行して、プール名を確認します。次に、`zpool status` コマンドを実行して、ブートディスクのデバイスパスを確認します。

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

        NAME                                STATE        READ WRITE CKSUM
        rpool                                ONLINE       0     0     0
        c0t5000CCA03C138904d0s0             ONLINE       0     0     0
```

3. システムのブートディスクに関する情報を取得します。

- Solaris I/O マルチパスによって管理されているディスクの場合、`mpathadm` コマンドを使用してブートディスクが接続されている接続されている PCIe バスを判断します。

SPARC T3 サーバーから、内部ディスクは、Solaris I/O マルチパスによって管理されています。

a. ディスクが接続されているイニシエータポートを見つけます。

```
primary# mpathadm show lu /dev/rdisk/c0t5000CCA03C138904d0s0
Logical Unit: /dev/rdisk/c0t5000CCA03C138904d0s2
mpath-support: libmpscsi_vhci.so
Vendor: HITACHI
Product: H106030SDSUN300G
Revision: A2B0
Name Type: unknown type
Name: 5000cca03c138904
Asymmetric: no
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
    Initiator Port Name: w50800200014100c8
    Target Port Name: w5000cca03c138905
    Override Path: NA
    Path State: OK
    Disabled: no

Target Ports:
    Name: w5000cca03c138905
    Relative ID: 0
```

b. イニシエータポートが存在する PCIe バスを判断します。

```
primary# mpathadm show initiator-port w50800200014100c8
Initiator Port: w50800200014100c8
Transport Type: unknown
OS Device File: /devices/pci@400/pci@2/pci@0/pci@e/scsi@0/iport@1
```

■ Solaris I/O マルチパスによって管理されていないディスクの場合、ls -l コマンドを使用してブロック型デバイスがリンクされている物理デバイスを判断します。

このコマンドは Solaris I/O マルチパスによって管理されていない UltraSPARC T2 または UltraSPARC T2 Plus システム上のディスクに対して使用します。

次の例では、ブロック型デバイス c1t0d0s0 を使用します。

```
primary# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root 49 Oct 1 10:39 /dev/dsk/c0t1d0s0 ->
../devices/pci@400/pci@0/pci@1/scsi@0/sd@1,0:a
```

この例では、primary ドメインのブートディスクに対する物理デバイスは、pci@400 に接続されています。

4. システムで使用されているネットワークインタフェースを確認します。

ifconfig コマンドを使用して、「plumb」されているプライマリネットワークインタフェースを確認します。plumb されたインタフェースには、IP プロトコルでデバイスを使用できるように、ストリームがセットアップされます。

```
primary# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
net0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 3
    inet 10.129.241.135 netmask ffffffff broadcast 10.129.241.255
    ether 0:10:e0:e:f1:78
```

```
primary# dladm show-phys net0
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up       1000   full     igb0
```

5. ネットワークインタフェースが接続されている物理デバイスを確認します。

次のコマンドでは、igb0 ネットワークインタフェースを使用します。

```
primary# ls -l /dev/igb0
lrwxrwxrwx 1 root root 46 Oct 1 10:39 /dev/igb0 ->
../devices/pci@500/pci@0/pci@c/network@0:igb0
```

ls -l /dev/usbecm コマンドも実行します。

この例では、primary ドメインによって使用されるネットワークインタフェースの物理デバイスは、前述の一覧の pci_1 に対応するバス pci@500 に接続しています。そのため、ほかの 2 つのバス pci_2 (pci@600) と pci_3 (pci@700) は primary ドメインでは使用されていないため、ほかのドメインに安全に割り当てることができます。

primary ドメインで使用されているネットワークインタフェースが、別のドメインに割り当てようとしているバス上にある場合は、別のネットワークインタフェースを使用するように primary ドメインを再構成します。

6. ブートディスクまたはネットワークインタフェースを含まないバスを primary ドメインから削除します。

この例では、pci_2 バスが primary ドメインから削除されます。

■ **動的方法:**

pci_2 バスのデバイスが primary ドメインの OS によって使用されていないことを確認します。使用されている場合、このコマンドはバスの削除に失敗する可能性があります。静的方法を使用して pci_2 バスを強制的に削除します。

```
primary# ldm remove-io pci_2 primary
```

■ **静的方法:**

バスを削除する前に、遅延再構成を開始する必要があります。

```
primary# ldm start-reconf primary
primary# ldm remove-io pci_2 primary
primary# shutdown -y -g0 -i6
```

primary ドメインでブートディスクおよびネットワークデバイスに使用されているバスは、ほかのドメインに割り当ててすることはできません。ほかの任意のバスを別のドメインに割り当てることができます。この例では、pci@600 は primary ドメインで使用されていないため、別のドメインに再割り当てすることができます。

7. バスをドメインに追加します。

この例では、pci_2 バスを ldg1 ドメインに追加します。

■ **動的方法:**

```
primary# ldm add-io pci_2 ldg1
```

■ **静的方法:**

バスを追加する前に、ドメインを停止する必要があります。

```
primary# ldm stop-domain ldg1
primary# ldm add-io pci_2 ldg1
primary# ldm start-domain ldg1
```

8. この構成をサービスプロセッサに保存します。

この例では、構成は io-domain です。

```
primary# ldm add-config io-domain
```

この構成 io-domain は、リブート後に使用される次の構成としても設定されます。

9. 適切なバスが primary ドメインに割り当てられたままで、適切なバスがドメイン ldg1 に割り当てられていることを確認します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	
pci_1	BUS	pci_1	primary	
pci_2	BUS	pci_2	ldg1	
pci_3	BUS	pci_3	primary	
/SYS/MB/PCIE1	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA0	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE5	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE6	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE7	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE2	PCIE	pci_2	ldg1	EMP
/SYS/MB/PCIE3	PCIE	pci_2	ldg1	EMP
/SYS/MB/PCIE4	PCIE	pci_2	ldg1	EMP
/SYS/MB/PCIE8	PCIE	pci_3	primary	EMP
/SYS/MB/SASHBA1	PCIE	pci_3	primary	OCC
/SYS/MB/NET2	PCIE	pci_3	primary	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_3	primary	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_3	primary	

この出力では、PCIe バス pci_0、pci_1、および pci_3 とそれらのデバイスが primary ドメインに割り当てられていることが確認されます。また、PCIe バス pci_2 とそのデバイスが ldg1 ドメインに割り当てられていることも確認されます。

PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成

この章では、PCIe SR-IOV に関する次のトピックについて説明します。

- [73 ページの「SR-IOV の概要」](#)
- [76 ページの「SR-IOV のハードウェア要件とソフトウェア要件」](#)
- [80 ページの「SR-IOV 機能の現在の制限事項」](#)
- [81 ページの「静的 SR-IOV」](#)
- [83 ページの「動的 SR-IOV」](#)
- [84 ページの「I/O 仮想化の有効化」](#)
- [85 ページの「PCIe SR-IOV 仮想機能の使用の計画」](#)
- [87 ページの「Ethernet SR-IOV 仮想機能の使用」](#)
- [107 ページの「InfiniBand SR-IOV 仮想機能の使用」](#)
- [121 ページの「ファイバチャネル SR-IOV 仮想機能の使用」](#)
- [136 ページの「I/O ドメインの回復性」](#)
- [144 ページの「非信頼性 I/O ドメインが構成された状態のルートドメインのリポート」](#)

SR-IOV の概要

注記 - ルートドメインはその他のルートドメインとの依存関係を持つことができないため、PCIe バスを所有するルートドメインは、その PCIe エンドポイントデバイスまたは SR-IOV 仮想機能をほかのルートドメインに割り当てることができません。ただし、PCIe バスから、そのバスを所有するルートドメインに、PCIe エンドポイントデバイスまたは仮想機能を割り当てることができます。

Peripheral Component Interconnect Express (PCIe) シングルルート I/O 仮想化 (SR-IOV) 実装は、PCI-SIG に定義されているバージョン 1.1 の規格に基づいています。SR-IOV

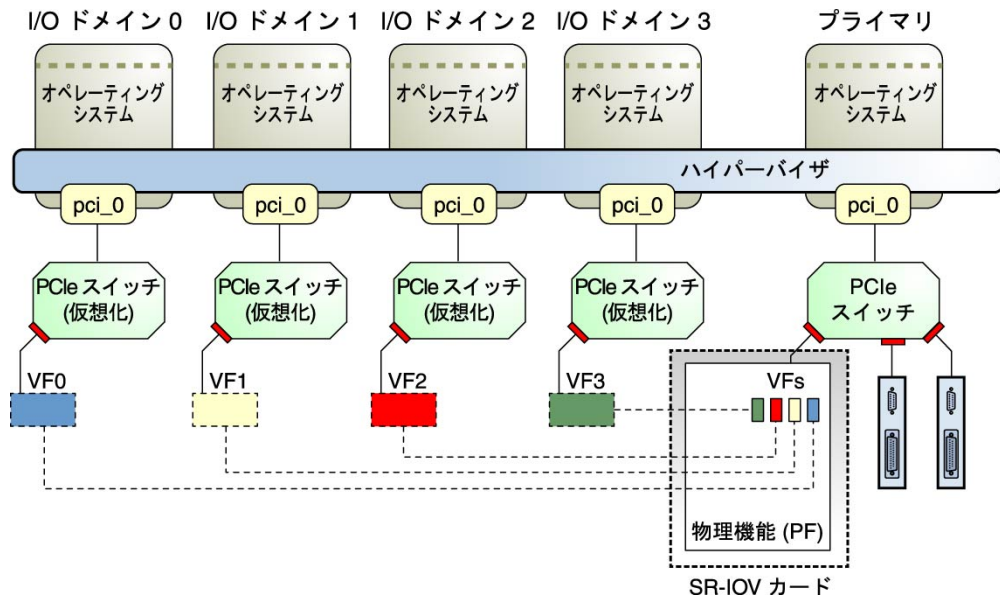
標準は、仮想マシン間での PCIe デバイスの効率的な共有を可能にし、ネイティブのパフォーマンスに匹敵する I/O パフォーマンスを達成するためにハードウェアに実装されます。SR-IOV 仕様は、新しいデバイスを作成し、それにより仮想マシンを I/O デバイスに直接接続できるという、新しい標準を定義しています。

物理機能として知られている 1 つの I/O リソースを、多数の仮想マシンで共有することができます。共有デバイスは専用のリソースを提供し、共有の共通リソースも使用します。このようにして、各仮想マシンが固有のリソースにアクセスすることができます。そのため、ハードウェアと OS が適切にサポートされる SR-IOV 対応の Ethernet ポートなどの PCIe デバイスは、それぞれに独自の PCIe 構成スペースがある、複数の別個の物理デバイスのように見えます。

SR-IOV の詳細は、PCI-SIG web site (<http://www.pcisig.com/>)を参照してください。

次の図は、I/O ドメイン内の仮想機能と物理機能の関係を示しています。

図 7-1 I/O ドメインでの仮想機能と物理機能の使用



SR-IOV には、次の機能タイプがあります。

- **物理機能** – SR-IOV 仕様によって定義されている SR-IOV 機能をサポートする PCI 機能。物理機能には SR-IOV 機能構造が含まれ、SR-IOV 機能を管理します。物理機能は

他の PCIe デバイスと同じように検出、管理、および操作が可能な多機能の PCIe 機能です。物理機能を使用して、PCIe デバイスを構成および制御できます。

- **仮想機能** – 物理機能に関連付けられた PCI 機能。仮想機能は軽量の PCIe 機能であり、1 つ以上の物理リソースを物理機能と共有したり、その物理機能に関連付けられた仮想機能と共有したりします。物理機能とは異なり、仮想機能はそれ自体の動作しか構成できません。

各 SR-IOV デバイスには 1 つの物理機能があり、各物理機能には最大 256 の仮想機能を関連付けることができます。この数は、特定の SR-IOV デバイスによって異なります。仮想機能は物理機能によって作成されます。

物理機能で SR-IOV が有効になった後、物理機能のバス、デバイス、および機能数によって、各仮想機能の PCI 構成スペースにアクセスすることができます。それぞれの仮想機能には、そのレジスタセットのマップに使用される PCI メモリー領域があります。仮想機能のデバイスドライバは、その機能を有効にするためにレジスタセットで動作し、仮想機能は実際の PCI デバイスのように見えます。作成した後、仮想機能を I/O ドメインに直接割り当てることができます。この機能により、仮想機能で物理デバイスを共有したり、CPU やハイパーバイザソフトウェアのオーバーヘッドなしで I/O を実行したりできます。

使用中の環境で SR-IOV 機能を使用すると、次のような利点が得られることがあります。

- **パフォーマンスの向上と待ち時間の削減** – 仮想マシン環境からハードウェアへの直接アクセス
- **コストの削減** – 次のような資産および運用上の支出の節約。
 - 省電力
 - アダプタ数の削減
 - 配線の削減
 - スイッチポートの削減

Oracle VM Server for SPARC SR-IOV の実装には、静的構成方法と動的構成方法の両方が含まれます。詳細については、[81 ページの「静的 SR-IOV」](#)および[83 ページの「動的 SR-IOV」](#)を参照してください。

Oracle VM Server for SPARC SR-IOV 機能により、次の操作を実行できます。

- 指定した物理機能での仮想機能の作成
- 物理機能の指定した仮想機能の破棄
- 仮想機能のドメインへの割り当て
- ドメインからの仮想機能の削除

SR-IOV 物理機能デバイスで仮想機能を作成および破棄するには、まず該当する PCIe バスで I/O 仮想化を有効にする必要があります。ldm set-io または ldm add-io コマンドを使用すると、iov プロパティを on に設定できます。また、ldm add-domain または ldm set-domain コマンドを使用すると、rc-add-policy プロパティを iov に設定できます。[ldm\(1M\)](#) マニュアルページを参照してください。

注記 - SPARC M7 シリーズサーバー、SPARC T7 シリーズサーバー、および Fujitsu M10 サーバーでは、PCIe バスの I/O 仮想化がデフォルトで有効になっています。

SR-IOV 仮想機能をドメインに割り当てると、SR-IOV 物理機能サービスを提供するドメインに対して暗黙的な依存関係が生まれます。ldm list-dependencies コマンドを使用することで、これらの依存関係を表示したり、この SR-IOV 物理機能に依存しているドメインを表示したりできます。[424 ページの「ドメイン I/O の依存関係の一覧表示」](#)を参照してください。

SR-IOV のハードウェア要件とソフトウェア要件

動的および静的 PCIe SR-IOV 機能は、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、および SPARC M7 シリーズサーバーでサポートされています。動的機能は、ほかのデバイスタイプで静的メソッドを使用する必要がある場合のみ、Fujitsu M10 プラットフォームの Ethernet デバイスでサポートされます。SPARC T3 プラットフォームでは、静的な PCIe SR-IOV 機能しかサポートされていません。

■ ハードウェア要件。

使用しているプラットフォームで利用できるカードを確認するには、プラットフォームのハードウェアドキュメントを参照してください。サポートされる PCIe カードの最新のリストについては、<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1> を参照してください。

- **Ethernet SR-IOV**。SR-IOV 機能を使用するには、オンボードの PCIe SR-IOV デバイスや PCIe SR-IOV 差し込み式カードを使用できます。プラットフォームのドキュメントに特に明記されていないかぎり、指定されたプラットフォームでのオンボードの SR-IOV デバイスはすべてサポートされています。
- **InfiniBand SR-IOV**。InfiniBand デバイスは、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでサポートされています。

- **ファイバチャネル SR-IOV**。ファイバチャネルデバイスは、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでサポートされています。

Fujitsu M10 プラットフォームでサポートされているデバイスの最新リストについては、お使いのモデルのプロダクトノートにある *Fujitsu M10/SPARC M10 Systems の PCI カードのインストールガイド* (<http://www.fujitsu.com/global/services/computing/server/sparc/downloads/manual/>) を参照してください。

- **ファームウェア要件**。

- **Ethernet SR-IOV**。動的 SR-IOV 機能を使用するには、SPARC T4 サーバーが少なくともバージョン 8.4.0.a のシステムファームウェアを実行している必要があります。SPARC T5 サーバー、SPARC M5 サーバー、および SPARC M6 サーバーは、少なくともバージョン 9.1.0.a のシステムファームウェアを実行している必要があります。SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーは、少なくともバージョン 9.4.3 のシステムファームウェアを実行している必要があります。Fujitsu M10 サーバーは、少なくともバージョン XCP2210 のシステムファームウェアを実行している必要があります。SPARC T3 サーバーは、静的 SR-IOV 機能のみをサポートしています。

SR-IOV 機能を使用するには、PCIe SR-IOV デバイスで、少なくともバージョン 3.01 のデバイスファームウェアが実行されている必要があります。Sun Dual 10-Gigabit Ethernet SFP+ PCIe 2.0 ネットワークアダプタのファームウェアを更新するには、次の手順を実行します。

1. デバイス上の FCode バージョンをアップグレードする必要があるかどうかを判断します。

ok プロンプトから、次のコマンドを実行します。

```
{0} ok cd path-to-device
{0} ok .properties
```

出力の version 値は、次のいずれかである必要があります。

LP	Sun Dual 10GbE SFP+ PCIe 2.0 LP FCode 3.01 4/2/2012
PEM	Sun Dual 10GbE SFP+ PCIe 2.0 EM FCode 3.01 4/2/2012

FEM Sun Dual 10GbE SFP+ PCIe 2.0 FEM FCode 3.01
4/2/2012

2. [My Oracle Support \(https://support.oracle.com/CSP/ui/flash.html#tab=PatchHomePage\(page=PatchHomePage&id=h0wvdx6\(\)\)\)](https://support.oracle.com/CSP/ui/flash.html#tab=PatchHomePage(page=PatchHomePage&id=h0wvdx6())) から、パッチ ID 13932765 をダウンロードします。

3. パッチをインストールします。

パッチパッケージには、ツールを使用してアップグレードを実行する方法を説明したドキュメントが含まれています。

- **InfiniBand SR-IOV**。この機能を使用するには、使用しているシステムで少なくとも次のバージョンのシステムファームウェアが実行されている必要があります。

- **SPARC T4** サーバー – 8.4
- **SPARC T5** サーバー – 9.1.0.x
- **SPARC T7** シリーズサーバー – 9.4.3
- **SPARC M5** および **SPARC M6** サーバー – 9.1.0.x
- **SPARC M7** シリーズサーバー – 9.4.3
- **Fujitsu M10** サーバー – XCP2210

InfiniBand SR-IOV デバイスとして Dual 40-Gigabit (4x) InfiniBand Host Channel Adapter M2 をサポートするには、そのカードまたはエクスプレスモジュールで少なくともバージョン 2.11.2010 のファームウェアが実行されている必要があります。このバージョンのファームウェアを入手するには、次のパッチをインストールします。

- **Low Profile (X4242A)** – パッチ ID 16340059
- **Express Module (X4243A)** – パッチ ID 16340042

primary ドメインのファームウェアを表示して更新する場合は、Oracle Solaris 11.1 の `fwflash` コマンドを使用します。現在のファームウェアバージョンを表示するには、`fwflash -lc IB` コマンドを使用します。ファームウェアを更新するには、`fwflash -f firmware-file -d device` コマンドを使用します。[fwflash\(1M\)](#) のマニュアルページを参照してください。

InfiniBand SR-IOV を使用するには、必ず InfiniBand スイッチでファームウェアバージョン 2.1.2 が実行されているようにします。このバージョンのファームウェアを入手するには、次のパッチをインストールします。

- **Sun Datacenter InfiniBand Switch 36 (X2821A-Z)** – パッチ ID 16221424

- **Sun Network QDR InfiniBand GatewaySwitch (X2826A-Z)** – パッチ ID 16221538

ファームウェアの更新方法については、使用している InfiniBand スイッチのドキュメントを参照してください。

- **ファイバチャネル SR-IOV**。この機能を使用するには、使用しているシステムで少なくとも次のバージョンのシステムファームウェアが実行されている必要があります。

- **SPARC T4** サーバー – 8.4.2.c
- **SPARC T5** サーバー – 9.1.2.d
- **SPARC T7** シリーズサーバー – 9.4.3
- **SPARC M5** サーバー – 9.1.2.d
- **SPARC M6** サーバー – 9.1.2.d
- **SPARC M7** シリーズサーバー – 9.4.3
- **Fujitsu M10** サーバー – XCP2210

ファイバチャネル SR-IOV 機能を有効にするには、Sun Storage 16 Gb Fibre Channel Universal HBA、Emulex 上のファームウェアのリビジョンが少なくとも 1.1.60.1 である必要があります。インストール手順は、ファームウェアに添付されています。



注意 - ファイバチャネル SR-IOV 機能を使用する予定の場合は、ファイバチャネルカードへのファームウェア更新のみを実行してください。

- **ソフトウェア要件。**

- **Ethernet SR-IOV**。SR-IOV 機能を使用するには、すべてのドメインが少なくとも Oracle Solaris 11.1 SRU 10 OS を実行している必要があります。
- **InfiniBand SR-IOV**。次のドメインで、サポートされている Oracle Solaris OS が動作している必要があります。
 - primary ドメインまたは primary 以外のルートドメインは、少なくとも Oracle Solaris 11.1 SRU 10 OS を実行している必要があります。
 - I/O ドメインは、少なくとも Oracle Solaris 11.1 SRU 10 OS を実行している必要があります。
 - 仮想機能の構成に使用する予定の InfiniBand SR-IOV 物理機能を備えたすべてのルートドメインで、`/etc/system` ファイルを更新します。


```
set ldc:ldc_maptable_entries = 0x20000
```

/etc/system プロパティ値の正しい作成または更新については、[400 ページの「/etc/system ファイルのプロパティ値の更新」](#)を参照してください。

仮想機能を追加する I/O ドメインで、/etc/system ファイルを更新します。

```
set rds3:rds3_fmr_pool_size = 16384
```

- **ファイバチャネル SR-IOV**。SR-IOV 機能を使用するには、すべてのドメインが少なくとも Oracle Solaris 11.1 SRU 17 OS を実行している必要があります。

静的および動的 SR-IOV ソフトウェア要件に関する詳細は、次を参照してください。

- [82 ページの「静的 SR-IOV ソフトウェア要件」](#)
- [83 ページの「動的 SR-IOV のソフトウェア要件」](#)

クラス固有の SR-IOV ハードウェア要件に関する詳細は、次を参照してください。

- [87 ページの「Ethernet SR-IOV のハードウェア要件」](#)
- [108 ページの「InfiniBand SR-IOV のハードウェア要件」](#)
- [122 ページの「ファイバチャネル SR-IOV のハードウェア要件」](#)

SR-IOV 機能の現在の制限事項

SR-IOV 機能には、次の制限事項があります。

- I/O ドメインは、関連するルートドメインが実行中でない場合は起動できません。
- 1 つ以上の仮想機能が割り当てられているドメインでは、移行は無効です。
- 破棄できるのは、物理機能用に最後に作成された仮想機能のみです。そのため、3 つの仮想機能を作成した場合、最初に破棄できる仮想機能は 3 番目の仮想機能です。
- 直接 I/O (DIO) 機能を使用して、ドメインに SR-IOV カードを割り当てた場合、そのカードに対して SR-IOV 機能は有効になりません。
- 特定の PCIe バスの PCIe エンドポイントデバイスおよび SR-IOV 仮想機能は、サポートされている SPARC T シリーズおよび SPARC M シリーズのシステムで最大 15 個のドメインに割り当てることができます。SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーでは、特定の PCIe バスの PCIe エンドポイントデバイスおよび SR-IOV 仮想機能を最大 31 個のドメインに割り当てることができます。Fujitsu M10 サーバーでは、特

定の PCIe バスの PCIe エンドポイントデバイスおよび SR-IOV 仮想機能を、最大 24 個のドメインに割り当てることができます。各 PCIe バスの割り込みベクトルなどの PCIe リソースは、ルートドメインと I/O ドメインの間で分配されます。その結果、特定の I/O ドメインに割り当てることができるデバイスの数も制限されています。同じ I/O ドメインに多数の仮想機能を割り当てないようにしてください。SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーに割り込みの制限はありません。SR-IOV に関連する問題については、『[Oracle VM Server for SPARC 3.3 リリースノート](#)』を参照してください。

- ルートドメインは PCIe バスの所有者で、バスの初期化と管理の役割を担います。ルートドメインはアクティブで、SR-IOV 機能をサポートするバージョンの Oracle Solaris OS を実行している必要があります。ルートドメインのシャットダウン、停止、またはリブートを行うと、PCIe バスへのアクセスが中断されます。PCIe バスが使用できないとき、そのバス上の PCIe デバイスが影響を受け、使用不可能になることがあります。

PCIe SR-IOV 仮想機能を持つ I/O ドメインの実行中にルートドメインがリブートされた場合、I/O ドメインの動作は予測不能です。たとえば、リブート中またはリブート後に、PCIe エンドポイントデバイスを持つ I/O ドメインでパニックが発生することがあります。ルートドメインをリブートするときは、各ドメインを手動で停止/開始する必要があります。

I/O ドメインに回復性がある場合は、PCIe バスを所有するルートドメインが使用不可になっても操作を続行できます。[136 ページの「I/O ドメインの回復性」](#)を参照してください。

- SPARC T5 および SPARC M6 プラットフォームまでの SPARC システムでは、提供される割り込み数が制限されているので、Oracle Solaris は各デバイスが使用可能な割り込み数を制限します。デフォルトの制限は一般的なシステム構成のニーズに一致する必要がありますが、特定のシステム構成ではこの値の調整が必要になることがあります。[421 ページの「割り込み制限の調整」](#)を参照してください。

静的 SR-IOV

静的 SR-IOV 方法では、ルートドメインが遅延再構成中か、SR-IOV 操作の実行中に I/O ドメインが停止されている必要があります。ルートドメインの構成手順を完了したら、それをリブートする必要があります。Oracle VM Server for SPARC 3.1 ファームウェアがシステムにインストールされていないか、各ドメインにインストールされている OS のバージョンが動的 SR-IOV をサポートしていない場合、この方法を使用する必要があります。

SR-IOV 仮想機能を作成または破棄するには、まず、ルートドメインで遅延再構成を開始する必要があります。それから、1 つまたは複数の `ldm create-vf` コマンドおよび `ldm destroy-vf`

コマンドを実行して、仮想機能を構成できます。最後に、ルートドメインをリブートします。次のコマンドは、`primary` 以外のルートドメインで仮想機能を作成する方法を示しています。

```
primary# ldm start-reconf root-domain-name
primary# ldm create-vf pf-name
primary# ldm stop-domain -r root-domain-name

primary# shutdown -i6 -g0 -y
```

静的に仮想機能を追加したり、ゲストドメインから仮想機能を削除したりするには、まずゲストドメインを停止する必要があります。次に、`ldm add-io` コマンドおよび `ldm remove-io` コマンドを実行して、仮想機能を構成します。変更が完了したら、ドメインを起動します。次のコマンドは、この方法で仮想機能を割り当てる方法を示しています。

```
primary# ldm stop guest-domain
primary# ldm add-io vf-name guest-domain
primary# ldm start guest-domain
```

さらに、ゲストドメインではなく、ルートドメインに仮想機能を追加したり、仮想機能を削除したりすることもできます。SR-IOV 仮想機能をルートドメインに追加するか、ルートドメインから削除するには、まず、ルートドメインで遅延再構成を開始します。これにより、1 つまたは複数の `ldm add-io` および `ldm remove-io` コマンドを実行できます。最後に、ルートドメインをリブートします。

ドメインのダウンタイムを最小限に抑えるには、仮想機能を構成する前に事前に計画します。

注記 - InfiniBand SR-IOV デバイスは、静的 SR-IOV でのみサポートされます。

静的 SR-IOV ソフトウェア要件

静的 SR-IOV 機能は、少なくとも Oracle VM Server for SPARC 3.0 のソフトウェアとファームウェアでサポートされています。[『Oracle VM Server for SPARC 3.0 Release Notes』の「PCIe SR-IOV Hardware and Software Requirements」](#)を参照してください。

`ldm set-io` または `ldm add-io` コマンドを使用すると、`iov` プロパティを `on` に設定できます。また、`ldm add-domain` または `ldm set-domain` コマンドを使用すると、`rc-add-policy` プロパティを `iov` に設定できます。[ldm\(1M\)](#) マニュアルページを参照してください。

ルートドメインをリブートすると SR-IOV に影響を及ぼすため、直接 I/O の構成変更は、SR-IOV に関連するルートドメインの変更が最大限になり、ルートドメインのリブートが最小限になるように、十分に計画してください。

動的 SR-IOV

動的 SR-IOV 機能によって、次の静的 SR-IOV 要件が削除されます。

- ルートドメイン。ルートドメインで遅延再構成を開始し、仮想機能を作成するか破棄し、ルートドメインをリブートします
- I/O ドメイン。I/O ドメインを停止し、仮想機能を追加または削除して、I/O ドメインを起動します

動的 SR-IOV では、ルートドメインで遅延再構成を開始する必要なく、仮想機能を動的に作成または破棄できます。仮想機能は、ドメインを停止しなくても、I/O ドメインに動的に追加したり削除したりすることができます。Logical Domains Manager は Logical Domains エージェントおよび Oracle Solaris I/O 仮想化フレームワークと通信し、これらの変更を動的に有効にします。

動的 SR-IOV のソフトウェア要件

必要な PCIe SR-IOV ソフトウェアおよびファームウェアのバージョンについては、[76 ページの「SR-IOV のハードウェア要件とソフトウェア要件」](#)を参照してください。

注記 - システムが、動的 SR-IOV ソフトウェアおよびファームウェア要件を満たしていない場合は、静的 SR-IOV 方法を使用して、SR-IOV 関連タスクを実行する必要があります。[81 ページの「静的 SR-IOV」](#)を参照してください。

動的 SR-IOV 構成の要件

仮想機能を動的に作成したり、破棄したりするには、次の条件が満たされていることを確認します。

- 仮想機能の構成を開始する前に、PCIe バスに対して I/O 仮想化が有効にされています。
- ルートドメインおよび I/O ドメイン上で実行されている OS は、少なくとも Oracle Solaris 11.1 SRU 10 OS である必要があります。
- 物理機能デバイスが OS に構成されていないか、マルチパス構成内にあります。たとえば、Ethernet SR-IOV デバイスを unplumb したり、IPMP またはアグリゲーションで使用したりして、仮想機能を正常に作成または破棄できます。

仮想機能を作成または破棄する操作では、物理機能デバイスドライバのオンライン状態とオフライン状態を切り替える必要があります。マルチパス構成では、デバイスドライバのこれらの状態の切り替えができます。

- 仮想機能は、I/O ドメインから削除する前に、使用されていないか、マルチパス構成ではありません。たとえば、Ethernet SR-IOV 仮想機能を unplumb することも、IPMP 構成で使用しないこともできます。

注記 - 現在のマルチパス実装では仮想機能をサポートしていないため、Ethernet SR-IOV 仮想機能にアグリゲーションは使用できません。

I/O 仮想化の有効化

SR-IOV 仮想機能を構成する前に、ルートドメインが遅延再構成中の場合に、PCIe バスの I/O 仮想化を有効にする必要があります。ドメインをリブートし、この変更を有効にします。

注記 - SPARC M7 シリーズサーバー、SPARC T7 シリーズサーバー、および Fujitsu M10 サーバーでは、PCIe バスの I/O 仮想化がデフォルトで有効になっています。

▼ PCIe バスに対する I/O 仮想化を有効にする方法

この手順は、ルートコンプレックスあたり 1 回のみ実行する必要があります。ルートコンプレックスは、同じ SP 構成の一部として実行している必要があります。

1. ルートドメインで遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

2. PCIe バスに対する I/O 仮想化操作を有効にします。

この手順は、物理機能を持つバスに対して、I/O 仮想化がすでに有効にされていない場合にのみ実行します。

次のいずれかのコマンドを実行します。

- 指定した PCIe バスがすでにルートドメインに割り当てられている場合に、I/O 仮想化を有効にします。

```
primary# ldm set-io iov=on bus
```

- ルートドメインに PCIe バスを追加しているときに仮想化を有効にします。

```
primary# ldm add-io iov=on bus
```

3. ルートドメインをリブートします。

次のいずれかのコマンドを実行します。

- **primary** 以外のルートドメインをリブートします。

```
primary# ldm stop-domain -r root-domain
```

- **primary** ルートドメインをリブートします。

```
primary# shutdown -i6 -g0 -y
```

PCIe SR-IOV 仮想機能の使用の計画

構成で仮想機能を使用する方法を事前に計画し、決定します。現在および将来の構成のニーズを満たす SR-IOV デバイスの仮想機能を判断します。

静的方法を使用する必要がある I/O 仮想化をまだ有効にしていない場合、この手順と仮想機能を作成する手順を組み合わせます。これらの手順を組み合わせることによって、ルートドメインのリブートが 1 回だけで済みます。

動的 SR-IOV が使用できる場合でも、推奨される方法は、すべての仮想機能を一度に作成することです。仮想機能を I/O ドメインに割り当てたあとに、それらを動的に作成できないことがあるためです。

静的 SR-IOV の場合、計画によって、それぞれ I/O ドメインに影響する可能性のあるルートドメインのリブートを何回も実行することを避けることができます。

I/O ドメインの詳細は、[62 ページの「I/O ドメインを作成するための一般的なガイドライン」](#)を参照してください。

SR-IOV 仮想機能の構成と割り当てを行うには、次の一般的な手順を使用してください。

1. 使用しているシステムで使用できる PCIe SR-IOV 物理機能と、そのなかでニーズにもっとも合うものを判断します。

次のコマンドを使用して、必要な情報を確認します。

<code>ldm list-io</code>	使用可能な SR-IOV 物理機能デバイスを確認します。
<code>prtdiag -v</code>	使用可能な PCIe SR-IOV カードおよびオンボードのデバイスを確認します。
<code>ldm list-io -l pf-name</code>	デバイスがサポートする仮想機能の最大数など、指定した物理機能についての追加情報を確認します。
<code>ldm list-io -d pf-name</code>	デバイスによってサポートされる、デバイス固有のプロパティを確認します。 100 ページの「高度な SR-IOV のトピック: Ethernet SR-IOV」 を参照してください。

2. PCIe バスに対する I/O 仮想化操作を有効にします。

[84 ページの「PCIe バスに対する I/O 仮想化を有効にする方法」](#)を参照してください。

3. 指定した SR-IOV 物理機能で、必要な数の仮想機能を作成します。

物理機能に対応する仮想機能を作成するには、次のコマンドを使用します。

```
primary# ldm create-vf -n max pf-name
```

詳細については、[88 ページの「Ethernet SR-IOV 仮想機能を作成する方法」](#)、[108 ページの「InfiniBand 仮想機能を作成する方法」](#)、および[125 ページの「ファイバチャネル SR-IOV 仮想機能を作成する方法」](#)を参照してください。

4. `ldm add-config` コマンドを使用して、構成を SP に保存します。

詳細については、[98 ページの「Ethernet SR-IOV 仮想機能を I/O ドメインに追加する方法」](#)、[112 ページの「InfiniBand 仮想機能を I/O ドメインに追加する方法」](#)、および[133 ページの「ファイバチャネル SR-IOV 仮想機能を I/O ドメインに追加する方法」](#)を参照してください。

Ethernet SR-IOV 仮想機能の使用

静的 SR-IOV 方法と動的 SR-IOV 方法の両方を使用すると、Ethernet SR-IOV デバイスを管理できます。

Ethernet SR-IOV のハードウェア要件

必要な PCIe Ethernet SR-IOV ハードウェアについては、[76 ページの「SR-IOV のハードウェア要件とソフトウェア要件」](#)を参照してください。

Ethernet SR-IOV の制限事項

今回のリリースでは、Ethernet SR-IOV 機能に次の制限事項があります。

- pvid または vid プロパティのいずれかを設定することで、仮想機能の VLAN 構成を有効にすることができます。これらの仮想機能プロパティの両方を同時に設定することはできません。
- SR-IOV 仮想機能を仮想スイッチのバックエンドデバイスとして使用することはできません。

Ethernet SR-IOV 仮想機能の使用の計画

仮想機能を動的に作成する際に、物理機能でマルチパスを使用していないこと、またはそれらが plumb されていないことを確認してください。

マルチパスを使用できないか、物理機能を plumb する必要がある場合は、静的方法式を使用して、仮想機能を作成します。[81 ページの「静的 SR-IOV」](#)を参照してください。

Ethernet デバイス固有のプロパティとネットワーク固有のプロパティ

`ldm create-vf` コマンドを使用して、仮想機能のデバイス固有およびネットワーク固有のプロパティを設定します。`unicast-slots` プロパティはデバイスに固有です。`mac-addr`、`alt-mac-addr`、`mtu`、`pvid`、および `vid` プロパティはネットワークに固有です。

`mac-addr`、`alt-mac-addr`、および `mtu` ネットワーク固有プロパティは、遅延再構成中で、仮想機能が `primary` ドメインに割り当てられている場合にのみ変更できます。

仮想機能が次のように割り当てられている場合、これらのプロパティを変更しようとすると失敗します。

- 仮想機能がアクティブな I/O ドメインに割り当てられている場合: 変更は所有しているドメインがアクティブでないかバインドされている状態のときに行われる必要があるため、プロパティ変更要求は拒否されます。
- 仮想機能が `primary` 以外のドメインに割り当てられていて、遅延再構成がすでに有効になっている場合: プロパティ変更要求は失敗して、エラーメッセージが表示されます。

`pvid` および `vid` のネットワーク固有プロパティは変更可能で、制限はありません。

Ethernet 仮想機能の作成

このセクションでは、仮想機能を動的に作成および破棄する方法を説明します。これらのアクションを実行するために動的方法を使用できない場合は、仮想機能を作成または破棄する前に、ルートドメインで遅延再構成を開始します。

▼ Ethernet SR-IOV 仮想機能を作成する方法

この動的方法を使用できない場合は、代わりに静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。

1. 仮想機能デバイスを識別します。

```
primary# ldm list-io
```

仮想機能の名前には、PCIe SR-IOV カードまたはオンボードのデバイスの場所情報が含まれています。

2. 物理機能を持つバスで I/O 仮想化が有効になっていない場合は、有効にします。

この手順は、物理機能を持つバスに対して、I/O 仮想化がすでに有効にされていない場合にのみ実行します。

84 ページの「PCIe バスに対する I/O 仮想化を有効にする方法」を参照してください。

3. Ethernet 物理機能から単一の仮想機能または複数の仮想機能を動的または静的に作成します。

1 つ以上の仮想機能を作成したあとに、それをゲストドメインに割り当てることができます。

■ 動的方法:

- 物理機能から複数の仮想機能を一度にすべて作成するには、次のコマンドを使用します。

```
primary# ldm create-vf -n number | max pf-name
```

該当する物理機能に対応するすべての仮想機能を一度に作成するには、ldm create-vf -n max コマンドを使用します。



注意 - システムで Intel 10-Gbit Ethernet カードが使用されている場合は、各物理機能から作成する仮想機能を 31 個以下にすることでパフォーマンスを最大にします。

バス名または仮名を使用して、仮想機能を指定することができます。ただし、推奨される方法は、仮名を使用することです。

- 物理機能から 1 つの仮想機能を作成するには、次のコマンドを使用します。

```
ldm create-vf [mac-addr=num] [alt-mac-addr=[auto|num1,[auto|num2,...]]  
[pvid=pvid] [vid=vid1,vid2,...] [mtu=size] [name=value...] pf-name
```

注記 - MAC アドレスが明示的に割り当てられていない場合、ネットワークデバイスに自動的に割り当てられます。

物理機能に対応する仮想機能を 1 つ作成するには、このコマンドを使用します。
Ethernet クラス固有のプロパティ値を手動で指定することもできます。

注記 - 新しく作成した仮想機能は、OS が IOV デバイスをプロブしている間は即時使用できない場合があります。ldm list-io コマンドを使用して、親の仮想機能と子の仮想機能の「ステータス」列に INV 値があるかどうかを確認します。これらの列にこの値が存在する場合は、ldm list-io の出力で「ステータス」列に INV 値が表示されなくなるのを待ってから (約 45 秒)、物理機能や子の仮想機能を使用します。このステータスが解決しない場合は、デバイスに問題があります。

ルートドメインのリブート (primary のリブートを含む) 直後や、ldm create-vf または ldm destroy-vf コマンドの使用直後は、デバイスのステータスが INV である可能性があります。

■ 静的方法:

- a. 遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

- b. Ethernet 物理機能から単一の仮想機能または複数の仮想機能を作成します。
仮想機能を動的に作成するには、上記で示したものと同一コマンドを使用します。

- c. ルートドメインをリブートします。

■ primary 以外のルートドメインをリブートするには:

```
primary# ldm stop-domain -r root-domain
```

■ primary ルートドメインをリブートするには:

```
primary# shutdown -i6 -g0 -y
```

例 7-1 Ethernet 物理機能に関する情報の表示

この例は、/SYS/MB/NET0/IOVNET.PF0 物理機能に関する情報を示しています。

- この物理機能は、オンボードの NET0 ネットワークデバイスからのものです。
- 文字列 IOVNET は、物理機能がネットワーク SR-IOV デバイスであることを示します。

```
primary# ldm list-io
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
niu_0                              NIU   niu_0   primary
niu_1                              NIU   niu_1   primary
pci_0                              BUS   pci_0   primary
```

pci_1	BUS	pci_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE3	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE5	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE7	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE9	PCIE	pci_1	primary	EMP
/SYS/MB/NET2	PCIE	pci_1	primary	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	primary	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	primary	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	primary	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	primary	

次のコマンドは、指定された物理機能についての詳細を表示します。値 maxvfs は、デバイスがサポートする仮想機能の最大数を示しています。

```
primary# ldm list-io -l /SYS/MB/NET0/IOVNET.PF0
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
/SYS/MB/NET0/IOVNET.PF0            PF    pci_0  primary
[pci@400/pci@1/pci@0/pci@4/network@0]
maxvfs = 7
```

例 7-2 オプションのプロパティを設定しない Ethernet 仮想機能の動的な作成

次の例は、オプションのプロパティを設定せずに仮想機能を動的に作成します。この場合、ネットワーククラス仮想機能の MAC アドレスは自動的に割り当てられます。

pci_0 PCIe バスで、I/O 仮想化が有効にされていることを確認します。[84 ページの「PCIe バスに対する I/O 仮想化を有効にする方法」](#)を参照してください。

これで、ldm create-vf コマンドを使用して、/SYS/MB/NET0/IOVNET.PF0 物理機能から仮想機能を作成できます。

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0
```

例 7-3 Ethernet 仮想機能の動的な作成とプロパティの設定

この例では、mac-addr プロパティを 00:14:2f:f9:14:c0 に、vid プロパティを VLAN ID 2 および 3 に設定して、仮想機能を動的に作成します。

```
primary# ldm create-vf mac-addr=00:14:2f:f9:14:c0 vid=2,3 /SYS/MB/NET0/IOVNET.PF0
```

例 7-4 2つの代替 MAC アドレスがある Ethernet 仮想機能の動的な作成

この例では、2つの代替 MAC アドレスがある仮想機能を動的に作成します。1つの MAC アドレスは自動的に割り当てられ、もう1つは 00:14:2f:f9:14:c2 として明示的に指定します。

```
primary# ldm create-vf alt-mac-addr=auto,00:14:2f:f9:14:c2 /SYS/MB/NET0/IOVNET.PF0
```

例 7-5 オプションのプロパティを設定しない仮想機能の静的な作成

次の例では、オプションのプロパティを設定せずに仮想機能を静的に作成します。この場合、ネットワーククラス仮想機能の MAC アドレスは自動的に割り当てられます。

まず、primary ドメインで遅延再構成を開始してから、pci_0 PCIe バスで I/O 仮想化を有効にします。pci_0 バスはすでに primary ルートドメインに割り当てられているため、ldm set-io コマンドを使用して、I/O 仮想化を有効にします。

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

```
primary# ldm set-io iov=on pci_0
```

これで、ldm create-vf コマンドを使用して、/SYS/MB/NET0/IOVNET.PF0 物理機能から仮想機能を作成できます。

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0
```

最後に、primary ルートドメインをリブートして、変更を有効にします。

```
primary# shutdown -i6 -g0 -y
```

例 7-6 複数の SR-IOV Ethernet 仮想機能の作成

次のコマンドは、/SYS/MB/NET2/IOVNET.PF1 物理機能から4つの仮想機能を作成する方法を示しています。

```
primary# ldm create-vf -n 31 /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF2
...
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF30
```

ldm create-vf -n コマンドは、必要に応じて、デフォルトのプロパティ値で設定された複数の仮想機能を作成します。ldm set-io コマンドを使用すると、あとでデフォルト以外のプロパティ値を指定できます。

Ethernet 仮想機能の破棄

仮想機能が現在ドメインに割り当てられていない場合は、その仮想機能を破棄できます。仮想機能は作成した逆の順序でしか破棄できないため、破棄できるのは最後に作成した仮想機能だけです。結果として作成される構成は、物理機能ドライバによって検証されます。

▼ Ethernet SR-IOV 仮想機能を破棄する方法

この動的方法を使用できない場合は、代わりに静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。

1. 仮想機能デバイスを識別します。

```
primary# ldm list-io
```

2. 単一の仮想機能または複数の仮想機能を動的または静的に破棄します。

■ 動的方法:

- 物理機能から仮想機能の一部またはすべてを一度に破棄するには、次のコマンドを使用します。

```
primary# ldm destroy-vf -n number | max pf-name
```

該当する物理機能に対応するすべての仮想機能を一度に破棄するには、ldm destroy-vf -n max コマンドを使用します。

-n オプションへの引数として *number* を指定した場合は、仮想機能の最後の *number* が破棄されます。この操作を実行しても 1 つの物理機能デバイスドライバの状態しか遷移しないため、この方法を使用してください。

パス名または仮名を使用して、仮想機能を指定することができます。ただし、推奨される方法は、仮名を使用することです。

■ 指定した仮想機能を破棄するには:

```
primary# ldm destroy-vf vf-name
```

影響を受けるハードウェアや OS で遅延が発生しているために、影響を受ける物理機能と残りの子の仮想機能が即時使用できない可能性もあります。ldm list-io コマンドを使用して、親の仮想機能と子の仮想機能の「ステータス」列に INV 値があるかどうかを確認します。これらの列にこの値が存在する場合は、ldm list-io の出力で「ステータス」列に INV 値が表示されなくなるのを待ちます (約 45 秒)。この時点で、その物理機能や子の仮想機能を安全に使用できるようになります。このステータスが解決しない場合は、デバイスに問題があります。

ルートドメインのリブート (primary のリブートを含む) 直後や、ldm create-vf または ldm destroy-vf コマンドの使用直後は、デバイスのステータスが INV である可能性があります。

■ 静的方法:

a. 遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

b. 単一の仮想機能または複数の仮想機能を破棄します。

■ 指定した物理機能から仮想機能を一度にすべて破棄するには、次のコマンドを使用します。

```
primary# ldm destroy-vf -n number | max pf-name
```

パス名または仮名を使用して、仮想機能を指定することができます。ただし、推奨される方法は、仮名を使用することです。

■ 指定した仮想機能を破棄するには:

```
primary# ldm destroy-vf vf-name
```

c. ルートドメインをリブートします。

■ **primary 以外のルートドメインをリブートするには:**

```
primary# ldm stop-domain -r root-domain
```

■ **primary ルートドメインをリブートするには:**

```
primary# shutdown -i6 -g0 -y
```

例 7-7 Ethernet 仮想機能の破棄

この例は、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を動的に破棄する方法を示しています。

```
primary# ldm destroy-vf /SYS/MB/NET0/IOVNET.PF0.VF0
```

次の例は、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を静的に破棄する方法を示しています。

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

```
primary# ldm destroy-vf /SYS/MB/NET0/IOVNET.PF0.VF0
primary# shutdown -i6 -g0 -y
```

例 7-8 複数の Ethernet SR-IOV 仮想機能の破棄

この例は、/SYS/MB/NET2/IOVNET.PF1 物理機能からすべての仮想機能を破棄した結果を示しています。ldm list-io の出力には、物理機能が 7 つの仮想機能を持っていることが示されています。ldm destroy-vf コマンドは、すべての仮想機能を破棄します。最終的な ldm list-io の出力には、仮想機能は 1 つも残っていないことが示されています。

```
primary# ldm list-io
...
/SYS/MB/NET2/IOVNET.PF1          PF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF0      VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF1      VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF2      VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF3      VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF4      VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF5      VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF6      VF    pci_1
primary# ldm destroy-vf -n max /SYS/MB/NET2/IOVNET.PF1
primary# ldm list-io
...
/SYS/MB/NET2/IOVNET.PF1          PF    pci_1    ldg1
```

Ethernet SR-IOV 仮想機能の変更

`ldm set-io vf-name` コマンドは、プロパティ値を変更するか、新規プロパティを設定することで、仮想機能の現在の構成を変更します。このコマンドは、ネットワーク固有のプロパティとデバイス固有のプロパティの両方を変更できます。デバイス固有のプロパティについては、[100 ページの「高度な SR-IOV のトピック: Ethernet SR-IOV」](#)を参照してください。

この動的方法を使用できない場合は、静的方法を使用します。[81 ページの「静的 SR-IOV」](#)を参照してください。

`ldm set-io` コマンドで、次のプロパティを変更できます。

- `mac-addr`, `alt-mac-addr`s, および `mtu`

これらの仮想機能のプロパティを変更するには、仮想機能を所有するドメインを停止し、`ldm set-io` コマンドを使用して、プロパティ値を変更し、ドメインを起動します。

- `pvid` および `vid`

仮想機能がドメインに割り当てられているときは、これらのプロパティを動的に変更することができます。そうすると、アクティブな仮想機能のネットワークトラフィックが変更されることがあります。`pvid` プロパティを設定すると、透過的 VLAN が有効になります。VLAN ID を指定するように `vid` プロパティを設定すると、それらの指定された VLAN に対して VLAN トラフィックが許可されます。

- デバイス固有のプロパティ

`ldm list-io -d pf-name` コマンドを使用して、有効なデバイス固有のプロパティのリストを表示します。物理機能と仮想機能の両方に対して、これらのプロパティを変更することができます。デバイス固有のプロパティを変更するには、静的方法を使用する必要があります。[81 ページの「静的 SR-IOV」](#)を参照してください。デバイス固有のプロパティの詳細については、[100 ページの「高度な SR-IOV のトピック: Ethernet SR-IOV」](#)を参照してください。

▼ Ethernet SR-IOV 仮想機能のプロパティを変更する方法

1. 仮想機能デバイスを識別します。

```
primary# ldm list-io
```

仮想機能の名前には、PCIe SR-IOV カードまたはオンボードのデバイスの場所情報が含まれています。

2. 仮想機能のプロパティを変更します。

```
ldm set-io name=value [name=value...] vf-name
```

例 7-9 Ethernet 仮想機能のプロパティの変更

次の例では、ldm set-io コマンドを使用して、仮想機能にプロパティを設定する方法について説明します。

- 次の例では、指定した仮想機能 /SYS/MB/NET0/IOVNET.PF0.VF0 のプロパティを変更して、VLAN ID 2、3、4 の一部になるようにします。

```
primary# ldm set-io vid=2,3,4 /SYS/MB/NET0/IOVNET.PF0.VF0
```

このコマンドは、仮想機能の VLAN の関連付けを動的に変更することに注意してください。これらの VLAN を変更するには、適切な Oracle Solaris OS ネットワークコマンドを使用して、I/O ドメインの VLAN インタフェースを構成する必要があります。

- 次の例では、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能に対して pvid プロパティ値を 2 に設定し、仮想機能を透過的に VLAN 2 の一部にします。つまり、仮想機能は、タグ付けされたどの VLAN トラフィックも表示しません。

```
primary# ldm set-io pvid=2 /SYS/MB/NET0/IOVNET.PF0.VF0
```

- 次の例では、自動的に割り当てられた 3 つの代替 MAC アドレスを、仮想機能に割り当てます。これらの代替アドレスにより、仮想機能の最上部に Oracle Solaris 11 仮想ネットワークインタフェースカード (VNIC) を作成することができます。VNIC を使用するには、ドメインで Oracle Solaris 11 OS を実行する必要があることに注意してください。

注記 - このコマンドを実行する前に、仮想機能を所有するドメインを停止します。

```
primary# ldm set-io alt-mac-addr=auto,auto,auto /SYS/MB/NET0/IOVNET.PF0.VF0
```

- 次の例では、指定された仮想機能に対して、デバイス固有の unicast-slots プロパティを 12 に設定します。物理機能に対して有効なデバイス固有のプロパティを調べるには、ldm list-io -d pf-name コマンドを使用します。

```
primary# ldm set-io unicast-slots=12 /SYS/MB/NET0/IOVNET.PF0.VF0
```

All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

I/O ドメイン上の Ethernet SR-IOV 仮想機能の追加と削除

▼ Ethernet SR-IOV 仮想機能を I/O ドメインに追加する方法

仮想機能を動的に削除できない場合は、静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。

1. I/O ドメインに追加する仮想機能を識別します。

```
primary# ldm list-io
```

2. 仮想機能を動的または静的に追加します。

■ 仮想機能を動的に追加するには:

```
primary# ldm add-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*domain-name* は仮想機能を追加するドメインの名前を指定します。

ドメイン内の仮想機能のデバイスパス名は、`list-io -l` 出力に示されているパスです。

■ 仮想機能を静的に追加するには:

- a. 遅延再構成を開始してから、仮想機能を追加します。

```
primary# ldm start-reconf root-domain-name
primary# ldm add-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*domain-name* は仮想機能を追加するドメインの名前を指定します。指定されたゲストは、アクティブでないか停止している必要があります。

ドメイン内の仮想機能のデバイスパス名は、`list-io -l` 出力に示されているパスです。

- b. ルートドメインをリブートします。

■ primary 以外のルートドメインをリブートするには:

```
primary# ldm stop-domain -r root-domain
```

■ primary ルートドメインをリブートするには:

```
primary# shutdown -i6 -g0 -y
```

例 7-10 Ethernet 仮想機能の追加

この例は、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を ldg1 ドメインに動的に追加する方法を示しています。

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

仮想機能を動的に追加できない場合は、静的方法を使用します。

```
primary# ldm stop-domain ldg1
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
primary# ldm start-domain ldg1
```

▼ Ethernet 仮想 SR-IOV 機能を I/O ドメインから削除する方法

仮想機能を動的に削除できない場合は、静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。



注意 - 仮想機能をドメインから削除する前に、そのドメインのブートが必須ではないことを確認してください。

1. I/O ドメインから削除する仮想機能を識別します。

```
primary# ldm list-io
```

2. 仮想機能を動的または静的に削除します。

■ 仮想機能を動的に削除するには:

```
primary# ldm remove-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、デバイスの仮名を使用することです。*domain-name* は仮想機能を削除するドメインの名前を指定します。

■ 仮想機能を静的に削除するには:

a. I/O ドメインを停止します。

```
primary# ldm stop-domain domain-name
```

b. 仮想機能を削除します。

```
primary# ldm remove-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、デバイスの仮名を使用することです。*domain-name* は仮想機能を削除するドメインの名前を指定します。指定されたゲストは、アクティブでないか停止している必要があります。

c. I/O ドメインを起動します。

```
primary# ldm start-domain domain-name
```

例 7-11 Ethernet 仮想機能の動的な削除

この例は、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を ldg1 ドメインから動的に削除する方法を示しています。

```
primary# ldm remove-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

コマンドが正常に実行されると、仮想機能が ldg1 ドメインから削除されます。ldg1 が再起動されると、指定された仮想機能はそのドメインに表示されなくなります。

仮想機能を動的に削除できない場合は、静的方法を使用します。

```
primary# ldm stop-domain ldg1
primary# ldm remove-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
primary# ldm start-domain ldg1
```

高度な SR-IOV のトピック: Ethernet SR-IOV

このセクションでは、SR-IOV 仮想機能の使用に関するいくつかの高度なトピックについて説明します。

仮想機能のための高度なネットワーク構成

SR-IOV 仮想機能を使用する場合は、次の問題に注意してください。

- SR-IOV 仮想機能は、Logical Domains Manager によって割り当てられた Mac アドレスのみを使用できます。ほかの Oracle Solaris OS ネットワークコマンドで I/O ドメイン上の MAC アドレスを変更すると、コマンドは失敗するか、適切に機能しない可能性があります。

- 現在のところ、I/O ドメインでの SR-IOV ネットワーク仮想機能のリンクアグリゲーションはサポートされていません。リンクアグリゲーションを作成しようとすると、予想どおりに機能しない可能性があります。
- I/O サービスを作成して、I/O ドメインに割り当てることができます。これらの仮想 I/O サービスは、仮想機能の作成元と同じ物理機能に作成することができます。たとえば、仮想スイッチ用のネットワークバックエンドデバイスとしてオンボードの 1 Gbps のネットワークデバイス (`net0` または `igb0`) を使用でき、同じ物理機能デバイスから仮想機能を静的に作成することもできます。

SR-IOV 仮想機能の使用による I/O ドメインのブート

SR-IOV 仮想機能には、仮想機能を論理ドメインのブートデバイスとして使用する機能など、ほかのタイプの PCIe デバイスと同様の機能があります。たとえば、ネットワーク仮想機能を使用して、Oracle Solaris OS を I/O ドメインにインストールするためにネットワークを介してブートすることができます。

注記 - 仮想機能デバイスから Oracle Solaris OS をブートするときは、ロードされる Oracle Solaris OS が仮想機能デバイスをサポートしていることを確認してください。その場合は、計画どおりに残りのインストールを続行できます。

SR-IOV デバイス固有のプロパティ

SR-IOV 物理機能のデバイスドライバは、デバイス固有のプロパティをエクスポートできます。これらのプロパティを使用して、物理機能とその仮想機能の両方のリソース割り当てを調整することができます。プロパティについては、[igb\(7D\)](#) および [ixgbe\(7D\)](#) マニュアルページなどの、物理機能ドライバについてのマニュアルページを参照してください。

`ldm list-io -d` コマンドは、指定された物理機能デバイスドライバによってエクスポートされるデバイス固有のプロパティを表示します。各プロパティの情報には、その名前、簡単な説明、デフォルト値、最大値、および次の 1 つまたは複数のフラグが含まれます。

P	物理機能に適用されます
V	仮想機能に適用されます
R	読み取り専用または通知パラメータ専用

primary# `ldm list-io -d pf-name`

`ldm create-vf` または `ldm set-io` コマンドを使用して、物理機能または仮想機能に対して読み書きプロパティを設定します。デバイス固有のプロパティを設定する場合は、静的方法を使用する必要があります。[81 ページの「静的 SR-IOV」](#)を参照してください。

次の例は、オンボードの Intel 1 ギガビット SR-IOV デバイスによってエクスポートされる、デバイス固有のプロパティを示しています。

```
primary# ldm list-io -d /SYS/MB/NET0/IOVNET.PF0
Device-specific Parameters
-----
max-config-vfs
  Flags = PR
  Default = 7
  Descr = Max number of configurable VFs
max-vf-mtu
  Flags = VR
  Default = 9216
  Descr = Max MTU supported for a VF
max-vlans
  Flags = VR
  Default = 32
  Descr = Max number of VLAN filters supported
pvid-exclusive
  Flags = VR
  Default = 1
  Descr = Exclusive configuration of pvid required
unicast-slots
  Flags = PV
  Default = 0 Min = 0 Max = 24
  Descr = Number of unicast mac-address slots
```

次の例では、`unicast-slots` プロパティを 8 に設定します。

```
primary# ldm create-vf unicast-slots=8 /SYS/MB/NET0/IOVNET.PF0
```

SR-IOV 仮想機能での VNIC の作成

SR-IOV 仮想機能では Oracle Solaris 11 VNIC の作成がサポートされています。ただし、サポートされる VNIC の数は、仮想機能に割り当てられた代替 MAC アドレス (`alt-mac-addr` プロパティ) の数に制限されます。仮想機能で VNIC を使用するときは、必ず十分な数の代替 MAC アドレスを割り当ててください。`ldm create-vf` または `ldm set-io` コマンドを使用して、代替 MAC アドレスを指定して `alt-mac-addr` プロパティを設定します。

次の例は、SR-IOV 仮想機能での 4 つの VNIC の作成を示しています。最初のコマンドは、代替 MAC アドレスを仮想機能デバイスに割り当てます。このコマンドは、自動割り当て方式

で、4 つの代替 MAC アドレスを `/SYS/MB/NET0/IOVNET.PF0.VF0` 仮想機能デバイスに割り当てます。

```
primary# ldm set-io alt-mac-addr=auto,auto,auto,auto /SYS/MB/NET0/IOVNET.PF0.VF0
```

次のコマンドは `ldg1` I/O ドメインを起動します。この例で、`auto-boot?` プロパティが `true` に設定されているため、I/O ドメインで Oracle Solaris 11 OS もブートされます。

```
primary# ldm start ldg1
```

次のコマンドは、ゲストドメインで Oracle Solaris 11 `dladm` コマンドを使用して、代替 MAC アドレスを持つ仮想機能を表示します。この出力には、`net30` 仮想機能が 4 つの代替 MAC アドレスを持つことを示しています。

```
guest# dladm show-phys -m
LINK          SLOT    ADDRESS              INUSE  CLIENT
net0           primary  0:14:4f:fa:b4:d1    yes    net0
net25          primary  0:14:4f:fa:c9:eb    no     --
net30          primary  0:14:4f:fb:de:4c    no     --
               1      0:14:4f:f9:e8:73    no     --
               2      0:14:4f:f8:21:58    no     --
               3      0:14:4f:fa:9d:92    no     --
               4      0:14:4f:f9:8f:1d    no     --
```

次のコマンドは、4 つの VNIC を作成します。代替 MAC アドレスを使用して指定した VNIC よりも多くの VNIC を作成しようとすると、失敗することに注意してください。

```
guest# dladm create-vnic -l net30 vnic0
guest# dladm create-vnic -l net30 vnic1
guest# dladm create-vnic -l net30 vnic2
guest# dladm create-vnic -l net30 vnic3
guest# dladm show-link
LINK          CLASS    MTU    STATE    OVER
net0          phys     1500   up       --
net25         phys     1500   up       --
net30         phys     1500   up       --
vnic0         vnic     1500   up       net30
vnic1         vnic     1500   up       net30
vnic2         vnic     1500   up       net30
vnic3         vnic     1500   up       net30
```

I/O ドメインを作成するための SR-IOV 仮想機能の使用

次の手順で、PCIe SR-IOV 仮想機能を含む I/O ドメインを作成する方法を説明します。

▼ SR-IOV 仮想機能を割り当てることによって I/O ドメインを作成する方法

ルートドメインのリブート回数を最小限に抑えて、停止時間を最小限に抑えるように事前に計画してください。

始める前に 始める前に、仮想機能を作成する物理機能の親である PCIe バスの I/O 仮想化を有効化していることを確認します。[84 ページの「PCIe バスに対する I/O 仮想化を有効にする方法」](#)を参照してください。

1. SR-IOV 機能を使用する I/O ドメインと共有する SR-IOV 物理機能を識別します。

```
primary# ldm list-io
```

2. 物理機能に対して 1 つ以上の仮想機能を作成します。

```
primary# ldm create-vf pf-name
```

このコマンドは、作成する仮想機能ごとに実行できます。-n オプションを使用すると、1 回のコマンドで同じ物理機能から複数の仮想機能を作成することもできます。[例7-6「複数の SR-IOV Ethernet 仮想機能の作成」](#)および [ldm\(1M\)](#) のマニュアルページを参照してください。

注記 - このコマンドは、ほかの仮想機能が関連付けられている物理機能からすでに作成されている場合、およびそれらの仮想機能のいずれかが別のドメインにバインドされている場合に失敗します。

3. ルートドメインで使用できる仮想機能のリストを表示します。

```
primary# ldm list-io
```

4. [ステップ 2](#) で作成した仮想機能をそのターゲット I/O ドメインに割り当てます。

```
primary# ldm add-io vf-name domain-name
```

注記 - ターゲット I/O ドメインの OS で動的 SR-IOV をサポートしていない場合は、静的方法を使用する必要があります。[81 ページの「静的 SR-IOV」](#)を参照してください。

5. 仮想機能が I/O ドメインで使用可能であることを確認します。

次の Oracle Solaris 11 コマンドは、仮想機能の利用度を表示します。

```
guest# dladm show-phys
```


例 7-12 SR-IOV 仮想機能を割り当てることによる I/O ドメインの動的な作成

次の動的な例は、物理機能 /SYS/MB/NET0/IOVNET.PF0 に対する仮想機能 /SYS/MB/NET0/IOVNET.PF0.VF0 を作成して、仮想機能を ldg1 I/O ドメインに割り当てる方法を示しています。この例では、次の状況が当てはまることを前提としています。

- primary ドメインの OS が動的 SR-IOV 操作をサポートしている
- pci_0 バスが primary ドメインに割り当てられ、I/O 仮想化操作のために初期化されている
- /SYS/MB/NET0/IOVNET.PF0 物理機能が pci_0 バスに属している
- /SYS/MB/NET0/IOVNET.PF0 物理機能に、ドメインに割り当てられた既存の仮想機能がない
- ldg1 ドメインがアクティブでブートされており、その OS が動的 SR-IOV 操作をサポートしている

/SYS/MB/NET0/IOVNET.PF0 物理機能から仮想機能を作成します。

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0
```

/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を ldg1 ドメインに追加します。

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

次のコマンドは、ldg1 ドメインに仮想機能が追加されていることを示しています。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE3	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE5	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE7	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE9	PCIE	pci_1	primary	EMP
/SYS/MB/NET2	PCIE	pci_1	primary	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	primary	

```

/SYS/MB/PCIE5/IOVNET.PF1    PF    pci_1    primary
/SYS/MB/NET2/IOVNET.PF0    PF    pci_1    primary
/SYS/MB/NET2/IOVNET.PF1    PF    pci_1    primary
/SYS/MB/NET0/IOVNET.PF0.VF0 VF    pci_0    ldg1

```

例 7-13 SR-IOV 仮想機能を割り当てることによる I/O ドメインの静的な作成

次の静的な例は、物理機能 `/SYS/MB/NET0/IOVNET.PF0` に対する仮想機能 `/SYS/MB/NET0/IOVNET.PF0.VF0` を作成して、仮想機能を `ldg1` I/O ドメインに割り当てる方法を示しています。この例では、次の状況が当てはまることを前提としています。

- `primary` ドメインの OS が動的 SR-IOV 操作をサポートしていない
- `pci_0` バスが `primary` ドメインに割り当てられ、I/O 仮想化操作のために初期化されていない
- `/SYS/MB/NET0/IOVNET.PF0` 物理機能が `pci_0` バスに属している
- `/SYS/MB/NET0/IOVNET.PF0` 物理機能に、ドメインに割り当てられた既存の仮想機能がない
- `ldg1` ドメインがアクティブでブートされており、その OS が動的 SR-IOV 操作をサポートしていない
- `ldg1` ドメインの `auto-boot?` プロパティが `true` に設定されているため、ドメインが起動時に自動的にブートする

まず、`primary` ドメインで遅延再構成を開始し、I/O 仮想化を有効にして、`/SYS/MB/NET0/IOVNET.PF0` 物理機能から仮想機能を作成します。

```

primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.

primary# ldm set-io iov=on pci_0
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0

-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0

```

次に、`primary` ドメインを停止します。

```
primary# shutdown -i6 -g0 -y
```

`ldg1` ドメインを停止し、仮想機能を追加して、ドメインを起動します。

```
primary# ldm stop ldg1
```

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
primary# ldm start ldg1
```

次のコマンドは、ldg1 ドメインに仮想機能が追加されていることを示しています。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	----	-----	-----
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE3	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE5	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE7	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE9	PCIE	pci_1	primary	EMP
/SYS/MB/NET2	PCIE	pci_1	primary	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	primary	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	primary	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	primary	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	primary	
/SYS/MB/NET0/IOVNET.PF0.VF0	VF	pci_0	ldg1	

InfiniBand SR-IOV 仮想機能の使用

InfiniBand SR-IOV デバイスでは、静的 SR-IOV 機能のみがサポートされます。

停止時間を最小限に抑えるため、ルートドメインが遅延再構成中か、ゲストドメインが停止している間に、すべての SR-IOV コマンドをグループとして実行します。この方法で制限されている SR-IOV コマンドは、ldm create-vf、ldm destroy-vf、ldm add-io、および ldm remove-io コマンドです。

一般に、仮想機能は複数のゲストドメインに割り当てられます。ルートドメインのリポートは、ルートドメインの仮想機能が割り当てられているすべてのゲストドメインに影響を与えます。

使用されていない InfiniBand 仮想機能のオーバーヘッドはほとんどないため、すぐに使用しない場合でも、事前に必要な仮想機能を作成しておくことによって、停止時間を回避できます。

InfiniBand SR-IOV のハードウェア要件

必要な PCIe InfiniBand SR-IOV ハードウェアについては、[76 ページの「SR-IOV のハードウェア要件とソフトウェア要件」](#)を参照してください。

InfiniBand SR-IOV がサポートされるには、ルートドメインが少なくとも Oracle Solaris 11.1 SRU 10 OS を実行している必要があります。I/O ドメインは、少なくとも Oracle Solaris 11.1 SRU 10 OS を実行できます。

InfiniBand 仮想機能の作成および破棄

▼ InfiniBand 仮想機能を作成する方法

この手順では、InfiniBand SR-IOV 仮想機能を作成する方法について説明します。

1. ルートドメインで遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

2. **iov=on** を設定して I/O 仮想化を有効にします。

この手順は、物理機能を持つバスに対して、I/O 仮想化がすでに有効にされていない場合にのみ実行します。

```
primary# ldm set-io iov=on bus
```

3. そのルートドメインから物理機能に関連付けられている 1 つ以上の仮想機能を作成します。

```
primary# ldm create-vf pf-name
```

このコマンドは、作成する仮想機能ごとに実行できます。**-n** オプションを使用すると、1 回のコマンドで同じ物理機能から複数の仮想機能を作成することもできます。[例7-6「複数の SR-IOV Ethernet 仮想機能の作成」](#) および [ldm\(1M\)](#) のマニュアルページを参照してください。

4. ルートドメインをリブートします。

次のいずれかのコマンドを実行します。

- **primary** 以外のルートドメインをリブートします。

```
primary# ldm stop-domain -r root-domain
```

- **primary** ルートドメインをリブートします。

```
primary# shutdown -i6 -g0 -y
```

例 7-14 InfiniBand 仮想機能の作成

次の例に、/SYS/MB/RISER1/PCIE4/IOVIB.PF0 物理機能に関する情報を示します。

- この物理機能は PCIE スロット 4 にあります。
- 文字列 IOVIB は、物理機能が InfiniBand SR-IOV デバイスであることを示します。

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN  STATUS
----
pci_0                                  BUS   pci_0    primary
niu_0                                  NIU   niu_0    primary
/SYS/MB/RISER0/PCIE0                   PCIE   pci_0    primary EMP
/SYS/MB/RISER1/PCIE1                   PCIE   pci_0    primary EMP
/SYS/MB/RISER2/PCIE2                   PCIE   pci_0    primary EMP
/SYS/MB/RISER0/PCIE3                   PCIE   pci_0    primary OCC
/SYS/MB/RISER1/PCIE4                   PCIE   pci_0    primary OCC
/SYS/MB/RISER2/PCIE5                   PCIE   pci_0    primary EMP
/SYS/MB/SASHBA0                        PCIE   pci_0    primary OCC
/SYS/MB/SASHBA1                        PCIE   pci_0    primary OCC
/SYS/MB/NET0                           PCIE   pci_0    primary OCC
/SYS/MB/NET2                           PCIE   pci_0    primary OCC
/SYS/MB/RISER0/PCIE3/IOVIB.PF0         PF     pci_0    primary
/SYS/MB/RISER1/PCIE4/IOVIB.PF0         PF     pci_0    primary
/SYS/MB/NET0/IOVNET.PF0                PF     pci_0    primary
/SYS/MB/NET0/IOVNET.PF1                PF     pci_0    primary
/SYS/MB/NET2/IOVNET.PF0                PF     pci_0    primary
/SYS/MB/NET2/IOVNET.PF1                PF     pci_0    primary
```

次のコマンドは、指定された物理機能についての詳細を表示します。値 maxvfs は、デバイスがサポートする仮想機能の最大数を示しています。

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                     TYPE  BUS      DOMAIN  STATUS
----
/SYS/MB/RISER1/PCIE4/IOVIB.PF0         PF     pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64
```

次の例は、静的仮想機能を作成する方法を示しています。まず、primary ドメインで遅延再構成を開始し、pci_0 PCIe バスで I/O 仮想化を有効にします。pci_0 バスはすでに primary ルートドメインに割り当てられているため、ldm set-io コマンドを使用して、I/O 仮想化を有効にします。

```
primary# ldm start-reconf primary
```

Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

```
primary# ldm set-io iov=on pci_0
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

これで、ldm create-vf コマンドを使用して、/SYS/MB/RISER1/PCIE4/IOVIB.PF0 物理機能から仮想機能を作成できます。

```
primary# ldm create-vf /SYS/MB/RISER1/PCIE4/IOVIB.PF0
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
Created new vf: /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0
```

同じ遅延再構成中に複数の仮想機能を作成できます。次のコマンドは、2 つ目の仮想機能を作成します。

```
primary# ldm create-vf /SYS/MB/RISER1/PCIE4/IOVIB.PF0
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
Created new vf: /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1
```

最後に、primary ルートドメインをリブートして、変更を有効にします。

```
primary# shutdown -i6 -g0 -y
Shutdown started.
```

```
Changing to init state 6 - please wait
...
```

▼ InfiniBand 仮想機能を破棄する方法

この手順では、InfiniBand SR-IOV 仮想機能を破棄する方法について説明します。

仮想機能が現在ドメインに割り当てられていない場合は、その仮想機能を破棄できます。仮想機能は作成した逆の順序でしか破棄できないため、破棄できるのは最後に作成した仮想機能だけです。結果として作成される構成は、物理機能ドライバによって検証されます。

1. ルートドメインで遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

2. そのルートドメインから物理機能に関連付けられている 1 つ以上の仮想機能を破棄します。

```
primary# ldm destroy-vf vf-name
```

このコマンドは、破棄する仮想機能ごとに実行できます。-n オプションを使用すると、1 回のコマンドで同じ物理機能から複数の仮想機能を破棄することもできます。例7-8「複数の Ethernet SR-IOV 仮想機能の破棄」および [ldm\(1M\)](#) のマニュアルページを参照してください。

3. ルートドメインをリブートします。

次のいずれかのコマンドを実行します。

- primary 以外のルートドメインをリブートします。

```
primary# ldm stop-domain -r root-domain
```

- primary ルートドメインをリブートします。

```
primary# shutdown -i6 -g0 -y
```

例 7-15 InfiniBand 仮想機能の破棄

次の例に、静的 InfiniBand 仮想機能 /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1 を破棄する方法を示します。

ldm list-io コマンドは、バス、物理機能、および仮想機能に関する情報を表示します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	IOV
...				
/SYS/MB/RISER1/PCIE4/IOVIB.PF0	PF	pci_0	primary	
...				
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0	VF	pci_0		
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1	VF	pci_0		

ldm list-io -l コマンドを使用して、物理機能と関連する仮想機能の詳細を取得できます。

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
```

NAME	TYPE	BUS	DOMAIN	STATUS
...				
/SYS/MB/RISER1/PCIE4/IOVIB.PF0	PF	pci_0	primary	

[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]

```

maxvfs = 64
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0      VF      pci_0
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,1]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1      VF      pci_0
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,2]

```

仮想機能は、ドメインに割り当てられていない場合にのみ破棄できます。ldm list-io -l 出力の DOMAIN 列に、仮想機能が割り当てられているドメインの名前が表示されます。また、仮想機能は、作成した逆の順序で破棄する必要があります。そのため、この例では、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0 仮想機能を破棄する前に、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1 仮想機能を破棄する必要があります。

適切な機能を識別したら、それを破棄できます。まず、遅延再構成を開始します。

```

primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.

primary# ldm destroy-vf /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

```

遅延再構成中に 1 つ以上の ldm destroy-vf コマンドを発行できます。したがって、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0 も破棄できます。

最後に、primary ルートドメインをリブートして、変更を有効にします。

```

primary# shutdown -i6 -g0 -y
Shutdown started.

Changing to init state 6 - please wait
...

```

I/O ドメイン上の InfiniBand 仮想機能の追加および削除

▼ InfiniBand 仮想機能を I/O ドメインに追加する方法

この手順では、InfiniBand SR-IOV 仮想機能を I/O ドメインに追加する方法について説明します。

1. I/O ドメインを停止します。


```
primary# ldm stop-domain domain-name
```

2. I/O ドメインに、1 つ以上の仮想機能を追加します。

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*domain-name* は仮想機能を追加するドメインの名前を指定します。指定された I/O ドメインは、アクティブでない状態またはバインドされた状態である必要があります。

```
primary# ldm add-io vf-name domain-name
```

3. I/O ドメインを起動します。

```
primary# ldm start-domain domain-name
```

例 7-16 InfiniBand 仮想機能の追加

次の例に、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 仮想機能を iodom1 I/O ドメインに追加する方法を示します。

まず、割り当てる仮想機能を識別します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	----	-----	-----
pci_0	BUS	pci_0	primary	IOV
...				
/SYS/MB/RISER1/PCIE4/IOVIB.PF0	PF	pci_0	primary	
...				
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0	VF	pci_0		
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1	VF	pci_0		
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2	VF	pci_0		
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3	VF	pci_0		

I/O ドメインに仮想機能を追加するには、割り当て解除する必要があります。DOMAIN 列は、仮想機能が割り当てられているドメインの名前を示します。この例では、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 がドメインに割り当てられていません。

ドメインに仮想機能を追加するには、ドメインがアクティブでないかバインドされた状態である必要があります。

```
primary# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	NORM	UPTIME
primary	active	-n-cv-	UART	32	64G	0.2%	0.2%	56m
iodom1	active	-n----	5000	8	8G	33%	33%	25m

ldm list-domain 出力には、iodom1 I/O ドメインがアクティブであることが示されているため、それを停止する必要があります。

```
primary# ldm stop iodom1
LDom iodom1 stopped
primary# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  NORM  UPTIME
primary       active    -n-cv-   UART    32    64G     0.0%  0.0%  57m
iodom1        bound    ------ 5000     8     8G
```

これで、I/O ドメインに仮想機能を追加できます。

```
primary# ldm add-io /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 iodom1
primary# ldm list-io
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2      VF      pci_0      iodom1
```

I/O ドメインの停止中に 1 つ以上の仮想機能を追加できます。たとえば、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3 などのほかの未割り当ての仮想機能を iodom1 に追加できます。仮想機能を追加したら、I/O ドメインを再起動できます。

```
primary# ldm start iodom1
LDom iodom1 started
primary# ldm list-domain
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  NORM  UPTIME
primary       active    -n-cv-   UART    32    64G     1.0%  1.0%  1h 18m
iodom1        active    -n----   5000     8     8G     36%   36%  1m
```

▼ InfiniBand 仮想機能を I/O ドメインから削除する方法

この手順では、InfiniBand SR-IOV 仮想機能を I/O ドメインから削除する方法について説明します。

1. I/O ドメインを停止します。

```
primary# ldm stop-domain domain-name
```

2. I/O ドメインから、1 つ以上の仮想機能を削除します。

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、デバイスの仮名を使用することです。*domain-name* は仮想機能を削除するドメインの名前を指定します。指定された I/O ドメインは、アクティブでない状態またはバインドされた状態である必要があります。

注記 - 仮想機能を I/O ドメインから削除する前に、それがそのドメインのブートに必須ではないことを確認してください。

```
primary# ldm remove-io vf-name domain-name
```

3. I/O ドメインを起動します。

```
primary# ldm start-domain domain-name
```

例 7-17 InfiniBand 仮想機能の削除

次の例に、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 仮想機能を iodom1 I/O ドメインから削除する方法を示します。

まず、削除する仮想機能を識別します。

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN STATUS
----
pci_0                                  BUS   pci_0    primary IOV
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0        PF    pci_0    primary
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0    VF    pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1    VF    pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2    VF    pci_0    iodom1
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3    VF    pci_0    iodom1
```

DOMAIN 列に、仮想機能が割り当てられているドメインの名前が表示されます。/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 仮想機能は iodom1 に割り当てられています。

I/O ドメインから仮想機能を削除するには、ドメインがアクティブでないかバインドされた状態である必要があります。ldm list-domain コマンドを使用して、ドメインの状態を判断します。

```
primary# ldm list-domain
NAME      STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary   active  -n-cv-  UART  32    64G     0.3%  0.3%  29m
iodom1    active  -n----  5000  8      8G      17%   17%   11m
```

この例では、iodom1 ドメインがアクティブであるため、停止する必要があります。

```
primary# ldm stop iodom1
LDM iodom1 stopped
primary# ldm list-domain
NAME      STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary   active  -n-cv-  UART  32    64G     0.0%  0.0%  31m
iodom1    bound   ------  5000  8      8G
```

これで、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 仮想機能を iodom1 から削除できます。

```
primary# ldm remove-io /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 iodom1
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN STATUS
----
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2    VF    pci_0
...
```

仮想機能の DOMAIN 列は空になっています。

I/O ドメインの停止中に複数の仮想機能を削除できます。この例では、/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3 仮想機能も削除できます。仮想機能を削除したら、I/O ドメインを再起動できます。

```
primary# ldm start iodom1
LDom iodom1 started
primary# ldm list-domain
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	NORM	UPTIME
primary	active	-n-cv-	UART	32	64G	0.3%	0.3%	39m
iodom1	active	-n----	5000	8	8G	9.4%	9.4%	5s

ルートドメインへの InfiniBand 仮想機能の追加および削除

▼ InfiniBand 仮想機能をルートドメインに追加する方法

この手順では、InfiniBand SR-IOV 仮想機能をルートドメインに追加する方法について説明します。

1. 遅延再構成を開始します。

```
primary# ldm start-reconf root-domain
```

2. ルートドメインに、1 つ以上の仮想機能を追加します。

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*root-domain-name* は仮想機能を追加するルートドメインの名前を指定します。

```
primary# ldm add-io vf-name root-domain-name
```

3. ルートドメインをリブートします。

次のいずれかのコマンドを実行します。

- **primary 以外のルートドメインをリブートします。**

```
primary# ldm stop-domain -r root-domain-name
```

- **primary ルートドメインをリブートします。**

```
primary# shutdown -i6 -g0 -y
```

▼ InfiniBand 仮想機能をルートドメインから削除する方法

この手順では、InfiniBand SR-IOV 仮想機能をルートドメインから削除する方法について説明します。

1. 遅延再構成を開始します。

```
primary# ldm start-reconf root-domain
```

2. ルートドメインから、1 つ以上の仮想機能を削除します。

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*root-domain-name* は仮想機能を追加するルートドメインの名前を指定します。

```
primary# ldm remove-io vf-name root-domain-name
```

3. ルートドメインをリブートします。

次のいずれかのコマンドを実行します。

- **primary** 以外のルートドメインをリブートします。

```
primary# ldm stop-domain -r root-domain-name
```

- **primary** ルートドメインをリブートします。

```
primary# shutdown -i6 -g0 -y
```

高度な SR-IOV のトピック: InfiniBand SR-IOV

このセクションでは、InfiniBand 物理機能および仮想機能を識別し、Logical Domains Manager と InfiniBand 物理機能および仮想機能の Oracle Solaris ビューを関連付ける方法について説明します。

InfiniBand SR-IOV 仮想機能の一覧表示

次の例に、/SYS/MB/RISER1/PCIE4/IOVIB.PF0 物理機能に関する情報を表示するさまざまな方法を示します。文字列 IOVIB を含む物理機能名は、それが InfiniBand SR-IOV デバイスであることを示します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	----	-----	-----
pci_0	BUS	pci_0	primary	IOV
niu_0	NIU	niu_0	primary	
/SYS/MB/RISER0/PCIE0	PCIE	pci_0	primary	EMP
/SYS/MB/RISER1/PCIE1	PCIE	pci_0	primary	EMP
/SYS/MB/RISER2/PCIE2	PCIE	pci_0	primary	EMP
/SYS/MB/RISER0/PCIE3	PCIE	pci_0	primary	OCC
/SYS/MB/RISER1/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/RISER2/PCIE5	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA0	PCIE	pci_0	primary	OCC
/SYS/MB/SASHBA1	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/NET2	PCIE	pci_0	primary	OCC
/SYS/MB/RISER0/PCIE3/IOVIB.PF0	PF	pci_0	primary	
/SYS/MB/RISER1/PCIE4/IOVIB.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF0	VF	pci_0	primary	
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF1	VF	pci_0	primary	
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF2	VF	pci_0	iodom1	
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF3	VF	pci_0	iodom1	
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0	VF	pci_0	primary	
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1	VF	pci_0	primary	
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2	VF	pci_0	iodom1	
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3	VF	pci_0	iodom1	

ldm list-io -l コマンドは、指定した物理機能デバイス /SYS/MB/RISER1/PCIE4/IOVIB.PF0 に関する詳細を表示します。maxvfs 値は、物理デバイスでサポートされている仮想機能の最大数が 64 であることを示しています。物理機能に関連付けられた各仮想機能では、出力が次のように示されます。

- 機能名
- 機能のタイプ
- バス名
- ドメイン名
- 機能のオプションのステータス
- デバイスパス

この ldm list-io -l の出力には、VF0 および VF1 が primary ドメインに割り当てられており、VF2 および VF3 が iodom1 I/O ドメインに割り当てられていることが示されます。

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                TYPE  BUS  DOMAIN  STATUS
----
```

```
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0      VF      pci_0      primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,1]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1      VF      pci_0      primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,2]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2      VF      pci_0      iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,3]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3      VF      pci_0      iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,4]
```

InfiniBand SR-IOV 機能の識別

このセクションでは、InfiniBand SR-IOV デバイスを識別する方法について説明します。

ldm list-io -l コマンドを使用して、各物理機能および仮想機能に関連付けられている Oracle Solaris デバイスパス名を表示します。

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                     TYPE  BUS      DOMAIN  STATUS
----                                     -
/SYS/MB/RISER1/PCIE4/IOVIB.PF0          PF    pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0      VF    pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,1]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1      VF    pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,2]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2      VF    pci_0    iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,3]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3      VF    pci_0    iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,4]
```

各 IP over InfiniBand (IPoIB) インスタンスをその物理カードに一致させるには、dladm show-phys -L コマンドを使用します。たとえば、次のコマンドは、スロット PCIE4 にあるカード (前の ldm list-io -l の例で示された同じカード) を使用する IPoIB インスタンスを表示します。

```
primary# dladm show-phys -L | grep PCIE4
net5          ibp0          PCIE4/PORT1
net6          ibp1          PCIE4/PORT2
net19         ibp8          PCIE4/PORT1
net9          ibp9          PCIE4/PORT2
net18         ibp4          PCIE4/PORT1
net11         ibp5          PCIE4/PORT2
```

各 InfiniBand ホストチャネルアダプタ (HCA) デバイスは、グローバル一意識別子 (GUID) を持ちます。各ポートに GUID があります (通常は HCA への 2 つのポートがあります)。InfiniBand HCA の GUID は、アダプタを一意に識別します。ポート GUID は各 HCA ポー

トを一意に識別し、ネットワークデバイスの MAC アドレスに似た役割を持ちます。これらの 16 進数の GUID は、InfiniBand 管理ツールと診断ツールで使用されます。

InfiniBand SR-IOV デバイスに関する GUID 情報を取得するには、`dladm show-ib` コマンドを使用します。同じデバイスの物理機能と仮想機能は、関連した HCA GUID 値を持ちます。HCA GUID の 16 進数の 11 桁目は、物理機能とその仮想機能の関係を示します。HCAGUID 列と PORTGUID 列の先頭のゼロは表示されません。

たとえば、物理機能 PF0 には 2 つの仮想機能 VF0 および VF1 があり、これらは primary ドメインに割り当てられています。各仮想機能の 16 進数の 11 桁目は、関連する物理機能から 1 つ増やされます。そのため、PF0 の GUID が 8 の場合、VF0 と VF1 の GUID はそれぞれ 9 と A になります。

次の `dladm show-ib` コマンド出力は、net5 および net6 リンクが物理機能 PF0 に属することを示しています。net19 および net9 リンクは同じデバイスの VF0 に属し、net18 および net11 リンクは VF1 に属します。

```
primary# dladm show-ib
LINK          HCAGUID          PORTGUID          PORT STATE  PKEYS
net6          21280001A17F56  21280001A17F58   2    up    FFFF
net5          21280001A17F56  21280001A17F57   1    up    FFFF
net19         21290001A17F56  140500000000001  1    up    FFFF
net9          21290001A17F56  140500000000008  2    up    FFFF
net18         212A0001A17F56  140500000000002  1    up    FFFF
net11         212A0001A17F56  140500000000009  2    up    FFFF
```

次の `dladm show-phys` の出力にあるデバイスは、各リンクと基本となる InfiniBand ポートデバイス (ibpX) の関係を示しています。

```
primary# dladm show-phys
LINK          MEDIA          STATE          SPEED  DUPLEX  DEVICE
...
net6          Infiniband     up            32000  unknown ibp1
net5          Infiniband     up            32000  unknown ibp0
net19         Infiniband     up            32000  unknown ibp8
net9          Infiniband     up            32000  unknown ibp9
net18         Infiniband     up            32000  unknown ibp4
net11         Infiniband     up            32000  unknown ibp5
```

実際の InfiniBand ポート (IB ポート) デバイスパスを表示するには、`ls -l` コマンドを使用します。IB ポートデバイスは、`ldm list-io -l` 出力で表示されるデバイスパスの子です。物理機能は、`pciex15b3,673c@0` のような 1 部のユニットアドレスを持ち、仮想機能は 2 部のユニットアドレス `pciex15b3,1002@0,2` を持ちます。ユニットアドレスの 2 番目の部分は、仮想機能の数値より 1 つ大きくなります。(この例では、2 番目の成分は 2 であるため、このデバイスは仮想

機能 1 です。)次の出力は、`/dev/ibp0` が物理機能で、`/dev/ibp5` が仮想機能であることを示しています。

```
primary# ls -l /dev/ibp0
lrwxrwxrwx 1 root root 83 Apr 18 12:02 /dev/ibp0 ->
../devices/pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0/hermon@0/ibport@1,0,ipib:ibp0
primary# ls -l /dev/ibp5
lrwxrwxrwx 1 root root 85 Apr 22 23:29 /dev/ibp5 ->
../devices/pci@400/pci@1/pci@0/pci@0/pciex15b3,1002@0,2/hermon@3/ibport@2,0,ipib:ibp5
```

OpenFabrics `ibv_devices` コマンドを使用して、OpenFabrics デバイス名とノード (HCA) GUID を表示できます。仮想機能が存在する場合、Type 列は機能が物理か仮想かを示しています。

```
primary# ibv_devices
device          node GUID          type
-----
mlx4_4          0002c90300a38910  PF
mlx4_5          0021280001a17f56  PF
mlx4_0          0002cb0300a38910  VF
mlx4_1          0002ca0300a38910  VF
mlx4_2          00212a0001a17f56  VF
mlx4_3          0021290001a17f56  VF
```

ファイバチャネル SR-IOV 仮想機能の使用

SR-IOV ファイバチャネルホストバスアダプタ (HBA) には、1 つ以上のポートが搭載されている場合があります、それぞれが SR-IOV 物理機能として表示されます。ファイバチャネル物理機能は、そのデバイス名内の `IOVFC` 文字列で識別できます。

ファイバチャネル物理機能ごとに、カードの製造元から提供された一意のポート WWN (World Wide Name) およびノード WWN の値が割り当てられています。ファイバチャネル物理機能から仮想機能を作成すると、仮想機能はファイバチャネル HBA デバイスのように動作します。仮想機能ごとに、SAN ファブリックのポート WWN およびノード WWN で指定された一意の識別情報が割り当てられている必要があります。Logical Domains Manager を使用すると、ポート WWN およびノード WWN を自動または手動で割り当てることができます。独自の値を割り当てれば、どの仮想機能の識別情報でも完全に制御できます。

ファイバチャネル HBA 仮想機能では、SAN ファブリックにログインする際に、N_Port ID 仮想化 (NPIV) 方法が使用されます。この NPIV 要件のために、ファイバチャネル HBA ポートを NPIV 対応のファイバチャネルスイッチに接続する必要があります。仮想機能は、SR-IOV カードのハードウェアまたはファームウェアで完全に管理されます。これらの例外を除いて、ファイ

バチャネル仮想機能は SR-IOV ファイバ HBA 以外のデバイスと同じように動作します。SR-IOV 仮想機能は、SR-IOV 以外のデバイスと同じ機能を持っているため、どちらの構成でも、すべてのタイプの SAN ストレージデバイスがサポートされます。

仮想機能の一意のポート WWN およびノード WWN の値を使用すると、SAN 管理者は SR-IOV ファイバチャネル HBA ポート以外の場合と同じ方法で、仮想機能にストレージを割り当てることができます。この管理には、ゾーニング、LUN マスキング、および QoS (Quality of Service) が含まれます。ルートドメイン内の物理機能に表示させずに、特定の論理ドメインに排他的にアクセスできるようにストレージを構成できます。

静的 SR-IOV 方法と動的 SR-IOV 方法の両方を使用すると、ファイバチャネル SR-IOV デバイスを管理できます。

ファイバチャネル SR-IOV のハードウェア要件

必要な PCIe ファイバチャネル SR-IOV ハードウェアについては、[76 ページの「SR-IOV のハードウェア要件とソフトウェア要件」](#)を参照してください。

- 制御ドメイン。
 - **QLogic** カード。Oracle Solaris 11.2 OS 以上
 - **Emulex** カード。Oracle Solaris 11.1 SRU 17 OS 以上
- I/O ドメイン。
 - **QLogic** カード。Oracle Solaris 11.2 OS 以上
 - **Emulex** カード。Oracle Solaris 11.1 SRU 17 OS 以上

ファイバチャネル SR-IOV の要件と制限事項

ファイバチャネル SR-IOV 機能には、次のような推奨事項と制限事項があります。

- SR-IOV カードでは、SR-IOV 機能がサポートされている最新バージョンのファームウェアが実行されている必要があります。
- ファイバチャネル PCIe カードは、NPIV がサポートされていて、PCIe カードとの互換性のあるファイバチャネルスイッチに接続されている必要があります。
- Logical Domains Manager では、すべてのシステムの制御ドメインを同じ SAN ファブリックに接続し、同じマルチキャストドメインの一部にすることで、一意の `port-wwn` および `node-wwn` プロパティ値が適切に自動生成されます。

この環境を構成できない場合は、仮想機能を作成するときに、`node-wwn` および `port-wwn` の値を手動で指定する必要があります。この動作によって、命名競合は発生しなくなります。[123 ページの「ファイバチャネル仮想機能の World-Wide Name の割り当て」](#)を参照してください。

ファイバチャネルデバイスクラス固有のプロパティ

`ldm create-vf` または `ldm set-io` コマンドを使用すると、次のファイバチャネル仮想機能のプロパティを設定できます。

bw-percent ファイバチャネル仮想機能に割り当てられる帯域幅の割合を指定します。有効な値は、0 から 100 までです。ファイバチャネル物理機能の仮想機能に割り当てられた帯域幅の合計値が 100 を超えることはできません。その仮想機能が、同じ物理機能を共有するほかの仮想機能によってまだ予約されていない帯域幅の正当な割当量を受け取れるように、デフォルト値は 0 になっています。

node-wwn ファイバチャネル仮想機能のノードの World Wide Name (WWN) を指定します。ゼロ以外の値が有効です。デフォルトでは、この値は自動的に割り当てられます。この値を手動で指定する場合は、`port-wwn` プロパティの値も指定する必要があります。詳細については、[123 ページの「ファイバチャネル仮想機能の World-Wide Name の割り当て」](#)を参照してください。

port-wwn ファイバチャネル仮想機能のポートの WWN を指定します。ゼロ以外の値が有効です。デフォルトでは、この値は自動的に割り当てられます。この値を手動で指定する場合は、`node-wwn` プロパティの値も指定する必要があります。詳細については、[123 ページの「ファイバチャネル仮想機能の World-Wide Name の割り当て」](#)を参照してください。

`node-wwn` または `port-wwn` プロパティ値は、ファイバチャネル仮想機能が使用中のときには変更できません。ただし、`bw-percent` プロパティ値は、ファイバチャネル仮想機能が使用中のときでも動的に変更できます。

ファイバチャネル仮想機能の World-Wide Name の割り当て

Logical Domains Manager では、ファイバチャネル仮想機能の World-Wide Name の自動割り当てと手動割り当ての両方がサポートされています。

World-Wide Name の自動割り当て

Logical Domains Manager は、自動 MAC アドレス割り当てプールから一意の MAC アドレスを割り当て、IEEE 形式の `node-wwn` および `port-wwn` プロパティ値を作成します。

```
port-wwn = 10:00:XX:XX:XX:XX:XX:XX
node-wwn = 20:00:XX:XX:XX:XX:XX:XX
```

`XX:XX:XX:XX:XX:XX` は、自動的に割り当てられた MAC アドレスです。

同じファイバチャネルファブリックに接続されているすべてのシステムの制御ドメインが Ethernet でも接続されていて、同じマルチキャストドメインの一部になっている場合、この自動割り当て方法で一意の WWN が生成されます。この要件を満たすことができない場合は、一意の WWN を手動で割り当てる必要があります。これは、SAN で必須です。

World-Wide Name の手動割り当て

一意の WWN は、任意の方法を使用して構築できます。このセクションでは、Logical Domains Manager の手動 MAC アドレス割り当てプールから WWN を作成する方法について説明します。割り当てられた WWN の一意性を保証する必要があります。

Logical Domains Manager には、`00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF` の範囲で手動で割り当てられる 256,000 個の MAC アドレスのプールがあります。

次の例は、MAC アドレス `00:14:4F:FC:00:01` に基づいた `port-wwn` および `node-wwn` プロパティ値を示しています。

```
port-wwn = 10:00:00:14:4F:FC:00:01
node-wwn = 20:00:00:14:4F:FC:00:01
```

`00:14:4F:FC:00:01` は、手動で割り当てられた MAC アドレスです。MAC アドレスの自動割り当てについては、[261 ページの「自動または手動による MAC アドレスの割り当て」](#)を参照してください。

注記 - SAN ストレージの予測可能な構成を保証するには、WWN を手動で割り当てるのが最適な方法です。

すべてのシステムが Ethernet で同じマルチキャストドメインに接続されていない場合は、WWN の手動割り当て方法を使用する必要があります。この方法を使用すれば、ファイバチャネル仮想機能が破棄され、再作成されたときに、同じ WWN が使用されることを保証することもできます。

ファイバチャネル SR-IOV 仮想機能の作成

このセクションでは、仮想機能を動的に作成および破棄する方法を説明します。これらのアクションを実行するために動的方法を使用できない場合は、仮想機能を作成または破棄する前に、ルートドメインで遅延再構成を開始します。

▼ ファイバチャネル SR-IOV 仮想機能を作成する方法

この動的方法を使用できない場合は、代わりに静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。

1. 仮想機能デバイスを識別します。

```
primary# ldm list-io
```

仮想機能の名前には、PCIe SR-IOV カードまたはオンボードのデバイスの場所情報が含まれています。

2. 物理機能を持つバスで I/O 仮想化が有効になっていない場合は、有効にします。

この手順は、物理機能を持つバスに対して、I/O 仮想化がすでに有効にされていない場合にのみ実行します。

[84 ページの「PCIe バスに対する I/O 仮想化を有効にする方法」](#)を参照してください。

3. 物理機能から単一の仮想機能または複数の仮想機能を動的または静的に作成します。

1 つ以上の仮想機能を作成したあとに、それをゲストドメインに割り当てることができます。

■ 動的方法:

- 物理機能から複数の仮想機能を一度にすべて作成するには、次のコマンドを使用します。

```
primary# ldm create-vf -n number | max pf-name
```

該当する物理機能に対応するすべての仮想機能を一度に作成するには、`ldm create-vf -n max` コマンドを使用します。このコマンドは、仮想機能ごとにポート WWN とノード WWN を自動的に割り当て、`bw-percent` プロパティをデフォルト値の `0` に設定します。この値は、すべての仮想機能に公正な帯域幅が割り当てられるように指定します。

ヒント - 物理機能に対応するすべての仮想機能を一度に作成します。WWN を手動で割り当てる場合は、まず仮想機能をすべて作成してから、`ldm set-io` コマンドを使用して、仮想機能ごとに WWN の値を手動で割り当てます。この方法を使用すると、物理機能から仮想機能を作成するときの状態遷移の数が最小限になります。

パス名または仮名を使用して、仮想機能を指定することができます。ただし、推奨される方法は、仮名を使用することです。

- 物理機能から 1 つの仮想機能を作成するには、次のコマンドを使用します。

```
ldm create-vf [bw-percent=value] [port-wwn=value node-wwn=value] pf-name
```

ファイバチャネルクラス固有のプロパティ値を手動で指定することもできます。

注記 - 新しく作成した仮想機能は、OS が IOV デバイスをプローブしている間は即時使用できない場合があります。ldm list-io コマンドを使用して、親の仮想機能と子の仮想機能の「ステータス」列に INV 値があるかどうかを確認します。これらの列にこの値が存在する場合は、ldm list-io の出力で「ステータス」列に INV 値が表示されなくなるのを待ってから (約 45 秒)、物理機能や子の仮想機能を使用します。このステータスが解決しない場合は、デバイスに問題があります。

ルートドメインのリブート (primary のリブートを含む) 直後や、ldm create-vf または ldm destroy-vf コマンドの使用直後は、デバイスのステータスが INV である可能性があります。

- 静的方法:

- a. 遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

- b. 物理機能から単一の仮想機能または複数の仮想機能を作成します。

仮想機能を動的に作成するには、上記で示したものと同一コマンドを使用します。

- c. ルートドメインをリブートします。

- primary 以外のルートドメインをリブートするには:

```
primary# ldm stop-domain -r root-domain
```

- primary ルートドメインをリブートするには:

```
primary# shutdown -i6 -g0 -y
```

例 7-18 ファイバチャネル物理機能に関する情報の表示

この例は、/SYS/MB/PCIE7/IOVFC.PF0 物理機能に関する情報を示しています。

- この物理機能は、PCIE スロット PCIE7 の基板にあります。
- 文字列 IOVFC は、物理機能がファイバチャネル SR-IOV デバイスであることを示します。

```
primary# ldm list-io
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
pci_0                              BUS   pci_0   primary IOV
pci_1                              BUS   pci_1   rootdom1 IOV
niu_0                              NIU   niu_0   primary
niu_1                              NIU   niu_1   primary
/SYS/MB/PCIE0                      PCIE   pci_0   primary OCC
/SYS/MB/PCIE2                      PCIE   pci_0   primary OCC
/SYS/MB/PCIE4                      PCIE   pci_0   primary OCC
/SYS/MB/PCIE6                      PCIE   pci_0   primary EMP
/SYS/MB/PCIE8                      PCIE   pci_0   primary EMP
/SYS/MB/SASHBA                    PCIE   pci_0   primary OCC
/SYS/MB/NET0                      PCIE   pci_0   primary OCC
/SYS/MB/NET0/PCIE1                PCIE   pci_1   rootdom1 OCC
/SYS/MB/PCIE3                    PCIE   pci_1   rootdom1 OCC
/SYS/MB/PCIE5                    PCIE   pci_1   rootdom1 OCC
/SYS/MB/PCIE7                    PCIE   pci_1   rootdom1 OCC
/SYS/MB/PCIE9                    PCIE   pci_1   rootdom1 OCC
/SYS/MB/NET2                    PCIE   pci_1   rootdom1 OCC
/SYS/MB/NET0/IOVNET.PF0           PF     pci_0   primary
/SYS/MB/NET0/IOVNET.PF1           PF     pci_0   primary
/SYS/MB/PCIE5/IOVNET.PF0           PF     pci_1   rootdom1
/SYS/MB/PCIE5/IOVNET.PF1           PF     pci_1   rootdom1
/SYS/MB/PCIE7/IOVFC.PF0            PF     pci_1   rootdom1
/SYS/MB/PCIE7/IOVFC.PF1            PF     pci_1   rootdom1
/SYS/MB/NET2/IOVNET.PF0           PF     pci_1   rootdom1
/SYS/MB/NET2/IOVNET.PF1           PF     pci_1   rootdom1
```

次のコマンドは、指定された物理機能についての詳細を表示します。値 `maxvfs` は、デバイスがサポートする仮想機能の最大数を示しています。

```
primary# ldm list-io -l /SYS/MB/PCIE7/IOVFC.PF0
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
/SYS/MB/PCIE7/IOVNET.PF0           PF     pci_0   rootdom1
[pci@400/pci@1/pci@0/pci@6/SUNW,fcdev@0]
    maxvfs = 8
```

例 7-19 オプションのプロパティを設定しないファイバチャネル仮想機能の動的な作成

次の例は、オプションのプロパティを設定せずに仮想機能を動的に作成します。この場合、`ldm create-vf` コマンドは、デフォルトの帯域幅 (パーセント)、ポート WWN (World-Wide Name)、およびノード WWN の値を自動的に割り当てます。

`pci_1` PCIe バスで、I/O 仮想化が有効にされていることを確認します。[84 ページの「PCIe バスに対する I/O 仮想化を有効にする方法」](#)を参照してください。

`ldm create-vf` コマンドを使用すると、`/SYS/MB/PCIE7/IOVFC.PF0` 物理機能からすべての仮想機能を作成できます。

```
primary# ldm create-vf -n max /SYS/MB/PCIE7/IOVFC.PF0
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF0
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF1
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF2
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF3
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF4
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF5
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF6
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF7
```

例 7-20 ファイバチャネル仮想機能の動的な作成とプロパティの設定

この例では、bw-percent プロパティ値を 25 に設定したときに仮想機能を動的に作成し、ポート WWN とノード WWN を指定します。

```
primary# ldm create-vf port-wwn=10:00:00:14:4F:FC:00:01 \
node-wwn=20:00:00:14:4F:FC:00:01 bw-percent=25 /SYS/MB/PCIE7/IOVFC.PF0
```

例 7-21 オプションのプロパティを設定しないファイバチャネル仮想機能の静的な作成

次の例では、オプションのプロパティを設定せずに仮想機能を静的に作成します。この場合、ldm create-vf コマンドは、デフォルトの帯域幅 (パーセント)、ポート WWN (World-Wide Name)、およびノード WWN の値を自動的に割り当てます。

まず、rootdom1 ドメインで遅延再構成を開始します。次に、pci_1 PCIe バスで I/O 仮想化を有効にします。pci_1 バスはすでに rootdom1 ルートドメインに割り当てられているため、ldm set-io コマンドを使用して、I/O 仮想化を有効にします。

```
primary# ldm start-reconf rootdom1
Initiating a delayed reconfiguration operation on the rootdom1 domain.
All configuration changes for other domains are disabled until the rootdom1
domain reboots, at which time the new configuration for the rootdom1 domain
will also take effect.
```

```
primary# ldm set-io iov=on pci_1
```

ここで、ldm create-vf コマンドを使用すると、/SYS/MB/PCIE7/IOVFC.PF0 物理機能からすべての仮想機能を作成できます。

```
primary# ldm create-vf -n max /SYS/MB/PCIE7/IOVFC.PF0

-----
Notice: The rootdom1 domain is in the process of a delayed reconfiguration.
Any changes made to the rootdom1 domain will only take effect after it reboots.
-----

Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF0
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF1
```



```
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF2
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF3
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF4
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF5
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF6
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF7
```

最後に、次の方法のいずれかで、rootdom1 ルートドメインをリブートして、変更を有効にします。

- rootdom1 は、primary 以外のルートドメインです。

```
primary# ldm stop-domain -r rootdom1
```

- rootdom1 は、primary ドメインです

```
primary# shutdown -i6 -g0 -y
```

ファイバチャネル SR-IOV 仮想機能の破棄

仮想機能が現在ドメインに割り当てられていない場合は、その仮想機能を破棄できます。仮想機能は作成した逆の順序でしか破棄できないため、破棄できるのは最後に作成した仮想機能だけです。結果として作成される構成は、物理機能ドライバによって検証されます。

▼ ファイバチャネル SR-IOV 仮想機能を破棄する方法

この動的方法を使用できない場合は、代わりに静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。

1. 仮想機能デバイスを識別します。

```
primary# ldm list-io
```

2. 単一の仮想機能または複数の仮想機能を動的または静的に破棄します。

- 動的方法:

- 物理機能から複数の仮想機能を一度にすべて破棄するには、次のコマンドを使用します。

```
primary# ldm destroy-vf -n number | max pf-name
```

パス名または仮名を使用して、仮想機能を指定することができます。ただし、推奨される方法は、仮名を使用することです。

該当する物理機能に対応するすべての仮想機能を一度に破棄するには、`ldm destroy-vf -n max` コマンドを使用します。

`-n` オプションへの引数として *number* を指定した場合は、仮想機能の最後の *number* が破棄されます。この操作を実行しても 1 つの物理機能デバイスドライバの状態しか遷移しないため、この方法を使用してください。

■ 指定した仮想機能を破棄するには:

```
primary# ldm destroy-vf vf-name
```

影響を受けるハードウェアや OS で遅延が発生しているために、影響を受ける物理機能と残りの子の仮想機能が即時使用できない可能性があります。`ldm list-io` コマンドを使用して、親の仮想機能と子の仮想機能の「ステータス」列に `INV` 値があるかどうかを確認します。これらの列にこの値が存在する場合は、`ldm list-io` の出力で「ステータス」列に `INV` 値が表示されなくなるのを待ちます (約 45 秒)。この時点で、その物理機能や子の仮想機能を安全に使用できるようになります。このステータスが解決しない場合は、デバイスに問題があります。

ルートメインのリブート (primary のリブートを含む) 直後や、`ldm create-vf` または `ldm destroy-vf` コマンドの使用直後は、デバイスのステータスが `INV` である可能性があります。

■ 静的方法:

a. 遅延再構成を開始します。

```
primary# ldm start-reconf root-domain-name
```

b. 単一の仮想機能または複数の仮想機能を破棄します。

■ 指定した物理機能から仮想機能を一度にすべて破棄するには、次のコマンドを使用します。

```
primary# ldm destroy-vf -n number | max pf-name
```

パス名または仮名を使用して、仮想機能を指定することができます。ただし、推奨される方法は、仮名を使用することです。

■ 指定した仮想機能を破棄するには:

```
primary# ldm destroy-vf vf-name
```

c. ルートドメインをリブートします。

■ primary 以外のルートドメインをリブートするには:

```
primary# ldm stop-domain -r root-domain
```

■ primary ルートドメインをリブートするには:

```
primary# shutdown -i6 -g0 -y
```

例 7-22 複数のファイバチャネル SR-IOV 仮想機能の動的な破棄

この例は、/SYS/MB/PCIE5/IOVFC.PF1 物理機能からすべての仮想機能を破棄した結果を示しています。ldm list-io の出力には、物理機能が 8 つの仮想機能を持っていることが示されています。ldm destroy-vf -n max コマンドは、すべての仮想機能を破棄します。最終的な ldm list-io の出力には、仮想機能が 1 つも残っていないことが示されています。

```
primary# ldm list-io
...
/SYS/MB/PCIE5/IOVFC.PF1          PF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF0      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF1      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF2      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF3      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF4      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF5      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF6      VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF7      VF      pci_1
primary# ldm destroy-vf -n max /SYS/MB/PCIE5/IOVFC.PF1
primary# ldm list-io
...
/SYS/MB/PCIE5/IOVFC.PF1          PF      pci_1
```

例 7-23 ファイバチャネル仮想機能の破棄

この例は、/SYS/MB/PCIE7/IOVFC.PF0 物理機能から仮想機能を静的に破棄する方法を示しています。

```
primary# ldm start-reconf rootdom1
Initiating a delayed reconfiguration operation on the rootdom1 domain.
All configuration changes for other domains are disabled until the rootdom1
domain reboots, at which time the new configuration for the rootdom1 domain
will also take effect.
```

```
primary# ldm destroy-vf -n max /SYS/MB/PCIE7/IOVFC.PF0
```

```
primary# ldm stop-domain -r rootdom1
```

ファイバチャネル SR-IOV 仮想機能の変更

`ldm set-io` コマンドは、プロパティ値を変更するか、新規プロパティを設定することで、仮想機能の現在の構成を変更します。

この動的方法を使用できない場合は、静的方法を使用します。[81 ページの「静的 SR-IOV」](#)を参照してください。

`ldm set-io` コマンドを使用すると、`bw-percent`、`port-wwn`、および `node-wwn` プロパティを変更できます。

仮想機能がドメインに割り当てられているときは、`bw-percent` プロパティのみを動的に変更できます。

▼ ファイバチャネル SR-IOV 仮想機能のプロパティを変更する方法

1. 仮想機能デバイスを識別します。

```
primary# ldm list-io
```

仮想機能の名前には、PCIe SR-IOV カードまたはオンボードのデバイスの場所情報が含まれています。

2. 仮想機能のプロパティを変更します。

```
ldm set-io [bw-percent=value] [port-wwn=value node-wwn=value] pf-name
```

いつでも動的に変更できる `bw-percent` プロパティとは異なり、`port-wwn` および `node-wwn` プロパティ値は、仮想機能がドメインに割り当てられていないときにのみ動的に変更できません。

例 7-24 ファイバチャネル SR-IOV 仮想機能のプロパティの変更

この例では、帯域幅 (パーセント) およびポート WWN とノード WWN の値を指定するために、指定された仮想機能 `/SYS/MB/PCIE7/IOVFC.PF0.VF0` のプロパティを変更します。

```
primary# ldm set-io port-wwn=10:00:00:14:4f:fc:f4:7c \
node-wwn=20:00:00:14:4f:fc:f4:7c bw-percent=25 /SYS/MB/PCIE7/IOVFC.PF0.VF0
```

I/O ドメイン上のファイバチャネル SR-IOV 仮想機能の追加と削除

▼ ファイバチャネル SR-IOV 仮想機能を I/O ドメインに追加する方法

仮想機能を動的に削除できない場合は、静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。

1. I/O ドメインに追加する仮想機能を識別します。

```
primary# ldm list-io
```

2. 仮想機能を動的または静的に追加します。

■ 仮想機能を動的に追加するには:

```
primary# ldm add-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*domain-name* は仮想機能を追加するドメインの名前を指定します。

ドメイン内の仮想機能のデバイスパス名は、`list-io -l` 出力に示されているパスです。

■ 仮想機能を静的に追加するには:

- a. ドメインを停止してから、仮想機能を追加します。

```
primary# ldm stop-domain domain-name
primary# ldm add-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、仮名を使用することです。*domain-name* は仮想機能を追加するドメインの名前を指定します。指定されたゲストは、アクティブでないか停止している必要があります。

ドメイン内の仮想機能のデバイスパス名は、`list-io -l` 出力に示されているパスです。

- b. ドメインを再起動します。

```
primary# ldm start-domain domain-name
```

例 7-25 ファイバチャネル仮想機能の追加

この例は、/SYS/MB/PCIE7/IOVFC.PF0.VF0 仮想機能を ldg2 ドメインに動的に追加する方法を示しています。

```
primary# ldm add-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 ldg2
```

仮想機能を動的に追加できない場合は、静的方法を使用します。

```
primary# ldm stop-domain ldg2
primary# ldm add-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 ldg2
primary# ldm start-domain ldg2
```

▼ ファイバチャネル SR-IOV 仮想機能を I/O ドメインから削除する方法

この動的方法を使用できない場合は、代わりに静的方法を使用してください。[81 ページの「静的 SR-IOV」](#)を参照してください。



注意 - 仮想機能をドメインから削除する前に、そのドメインのブートが必須ではないことを確認してください。

1. I/O ドメインから削除する仮想機能を識別します。

```
primary# ldm list-io
```

2. 仮想機能を動的または静的に削除します。

■ 仮想機能を動的に削除するには:

```
primary# ldm remove-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、デバイスの仮名を使用することです。*domain-name* は仮想機能を削除するドメインの名前を指定します。

■ 仮想機能を静的に削除するには:

- a. I/O ドメインを停止します。

```
primary# ldm stop-domain domain-name
```

- b. 仮想機能を削除します。

```
primary# ldm remove-io vf-name domain-name
```

vf-name は、仮想機能の仮名またはパス名です。推奨される方法は、デバイスの仮名を使用することです。*domain-name* は仮想機能を削除するドメインの名前を指定します。指定されたゲストは、アクティブでないか停止している必要があります。

c. I/O ドメインを起動します。

```
primary# ldm start-domain domain-name
```

例 7-26 ファイバチャネル仮想機能の動的な削除

この例は、/SYS/MB/PCIE7/IOVFC.PF0.VF0 仮想機能を ldg2 ドメインから動的に削除する方法を示しています。

```
primary# ldm remove-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 ldg2
```

コマンドが正常に実行されると、仮想機能が ldg2 ドメインから削除されます。ldg2 が再起動されると、指定された仮想機能はそのドメインに表示されなくなります。

仮想機能を動的に削除できない場合は、静的方法を使用します。

```
primary# ldm stop-domain ldg2
primary# ldm remove-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 ldg2
primary# ldm start-domain ldg2
```

高度な SR-IOV のトピック: ファイバチャネル SR-IOV

このセクションでは、ファイバチャネル SR-IOV 仮想機能の使用に関するいくつかの高度なトピックについて説明します。

ゲストドメインのファイバチャネル仮想機能へのアクセス

ldg2 コンソールログには、割り当てられたファイバチャネル仮想機能デバイスの動作が表示されます。ファイバチャネル仮想機能デバイスの表示およびアクセスを行うには、`fcadm` コマンドを使用します。

```
ldg2# fcdm hba-port
HBA Port WWN: 100000144ffb8a99
Port Mode: Initiator
Port ID: 13d02
```

```
OS Device Name: /dev/cfg/c3
Manufacturer: Emulex
Model: 7101684
Firmware Version: 7101684 1.1.60.1
FCode/BIOS Version: Boot:1.1.60.1 Fcode:4.03a4
Serial Number: 4925382+133400002R
Driver Name: emlxs
Driver Version: 2.90.15.0 (2014.01.22.14.50)
Type: N-port
State: online
Supported Speeds: 4Gb 8Gb 16Gb
Current Speed: 16Gb
Node WWN: 200000144ffb8a99
NPIV Not Supported
```

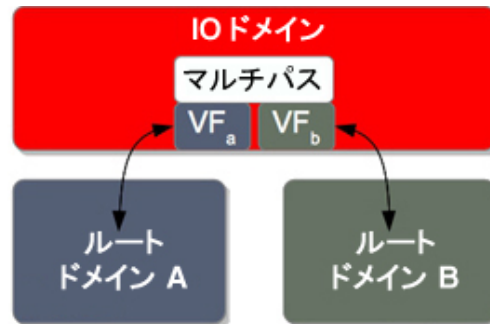
表示可能な LUN を表示するには、format コマンドを使用します。

```
ldg2# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-25.00GB>
    /virtual-devices@100/channel-devices@200/disk@0
  1. c3t21000024FF4C4BF8d0 <SUN-COMSTAR-1.0-10.00GB>
    /pci@340/pci@1/pci@0/pci@6/SUNW,emlxs@0,2/fp@0,0/ssd@w21000024ff4c4bf8,0
Specify disk (enter its number): ^D
ldg2#
```

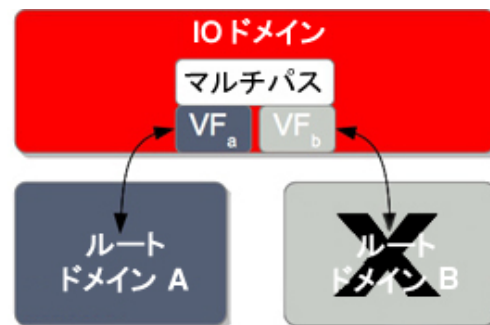
I/O ドメインの回復性

I/O ドメインの回復性は、関連付けられたルートドメインのいずれかが中断された場合でも I/O ドメインが続けて実行できるようにすることで、I/O ドメインの可用性とパフォーマンスを向上させます。ルートドメインが中断された場合は、影響を受けるデバイスを代替の I/O パスにフェイルオーバーすることで、そのサービスを使用する I/O ドメインが続けて実行できるようにします。ルートドメインがサービスに復帰すると、回復性 I/O ドメイン内の影響を受けたデバイスもサービスに復帰し、フェイルオーバー機能がリストアされます。

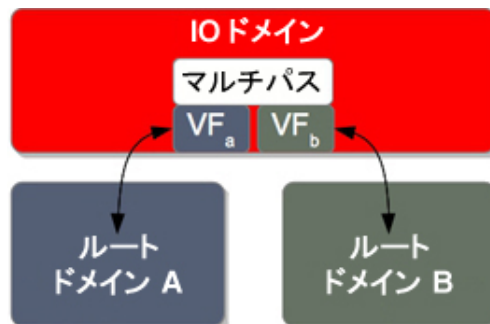
次の図は、構成されたルートドメインの 1 つに障害が起きたときに発生する動作と、ルートドメインがサービスに復帰したときに発生する動作について示しています。



各ルートドメインが I/O ドメインに仮想機能を 1 つずつ提供します。I/O ドメインは、ネットワークデバイス用の IPMP やファイバチャネルデバイス用の MPxIO などの仮想デバイスマルチパスを使用します。



パニックまたはリブートによってルートドメイン B が中断された場合、仮想機能 B が I/O ドメイン内で中断され、次にマルチパス機能がすべての I/O をルートドメイン A 経由でルーティングしようとしています。



ルートドメイン B がサービスに復帰すると、仮想機能 B が I/O ドメイン内で動作を再開します。マルチパスグループは完全な冗長性を取り戻します。

この構成では、仮想機能を仮想ネットワークデバイスまたは仮想ストレージデバイスにすることもできます。つまり、I/O ドメインは、仮想機能または仮想デバイスの任意の組み合わせで構成できます。

回復性と非回復性の両方の I/O ドメインを含む構成を作成できます。詳細は、[143 ページの「例 - 回復性および非回復性構成の使用」](#)を参照してください。

回復性 I/O ドメインの要件

注記 - Oracle Solaris 10 OS では、I/O ドメインの回復性は提供されません。

回復性 I/O ドメインは次の要件を満たす必要があります。

- 少なくとも Oracle Solaris 11.2 SRU 8 OS を実行し、その primary ドメインが少なくとも Oracle VM Server for SPARC 3.2 ソフトウェアを実行していること。
- マルチパス機能を使用して、仮想機能と仮想デバイスに対応するフェイルオーバー構成を作成すること。この構成では、仮想機能と仮想デバイスが同じクラス (ネットワークまたはストレージ) に属している必要があります。
- master プロパティの値には、failure-policy プロパティが ignore に設定されているルートドメインの名前を設定すること。その他の障害ポリシー設定 (stop、reset、panic など) では I/O の回復性が破棄され、I/O ドメインが中断されます。
- I/O ドメインの回復性をサポートする SR-IOV 仮想機能、仮想ネットワークデバイス、および仮想ストレージデバイスのみを使用すること。<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1> を参照してください。

I/O ドメインの回復性の制限事項

- SR-IOV カードをルートドメインにホットプラグして、I/O ドメインに仮想機能を割り当てた場合、ルートドメインに障害が発生したときに I/O ドメインが回復性を提供できない可能性があります。したがって、SR-IOV カードを追加するのはルートドメインが停止している間のみにする必要があります。次に、ルートドメインのブート後に仮想機能を割り当てます。
- 回復性 I/O ドメインがあり、デバイスを次のいずれかの方法で割り当てた場合、その I/O ドメインは回復性を失います。
 - I/O 回復性をサポートしないカードから仮想機能を追加した場合
 - 直接 I/O 機能を使用してデバイスを直接割り当てた場合

この場合、障害ポリシーを ignore から reset または stop に設定します。

回復性 I/O ドメインの構成

▼ 回復性 I/O ドメインを構成する方法

始める前に I/O ドメインの回復性機能をサポートする PCIe カードのみを使用します。<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1> を参照してください。

I/O ドメイン、ルートドメイン、サービสดメイン、および primary ドメインが少なくとも Oracle Solaris 11.2 SRU 8 OS および Logical Domains Manager 3.2 ソフトウェアを実行していることを確認してください。

1. ルートドメインで、**failure-policy** プロパティを **ignore** に設定します。

```
primary# ldm set-domain failure-policy=ignore root-domain-name
```

注記 - 回復性をサポートしないデバイスを I/O ドメインに追加すると、そのドメインは回復性を持たなくなります。そのため、**failure-policy** プロパティの値を **stop**、**reset**、または **panic** にリセットします。

ドメインの依存関係については、[406 ページの「ドメインの依存関係の構成」](#)を参照してください。

2. I/O ドメインで、**master** プロパティにルートドメインの名前を設定します。

```
primary# ldm set-domain master=root-domain-name I/O-domain-name
```

3. パス全体をマルチパス構成にします。

- Ethernet。IPMP を使用して、パス全体をマルチパス構成にします。

IPMP を使用したマルチパス構成については、[Administering TCP/IP Networks, IPMI, and IP Tunnels in Oracle Solaris 11.2](#)を参照してください。

- ファイバチャネル。MPxIO を使用して、パス全体をマルチパス構成にします。

MPxIO を使用したマルチパス構成については、[Managing SAN Devices and Multipathing Oracle Solaris 11.2](#)を参照してください。

例 7-27 IPMP を使用した Ethernet SR-IOV 機能によるマルチパス構成

この例は、IPMP を使用して、回復性 I/O ドメインに対応するネットワーク仮想機能デバイスを構成する方法を示しています。詳細は、[Administering TCP/IP Networks, IPMI, and IP Tunnels in Oracle Solaris 11.2](#)を参照してください。

1. 別々のルートドメインに割り当てられている 2 つの Ethernet SR-IOV 物理機能を特定します。

この例では、root-1 および root-2 ルートドメインに Ethernet SR-IOV 物理機能があります。

```
primary# ldm list-io | grep root-1 | grep PF
/SYS/PCI-EM8/IOVNET.PF0          PF      pci_1    root-1
primary# ldm list-io | grep root-2 | grep PF
/SYS/RIO/NET2/IOVNET.PF0        PF      pci_2    root-2
```

2. 2 つの Ethernet 仮想機能を、指定した物理機能のそれぞれに作成します。

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
Created new vf: /SYS/PCI-EM8/IOVNET.PF0.VF0
primary# ldm create-vf /SYS/RIO/NET2/IOVNET.PF0
Created new vf: /SYS/RIO/NET2/IOVNET.PF0.VF0
```

3. Ethernet 仮想機能を io-1 I/O ドメインに割り当てます。

```
primary# ldm add-io /SYS/PCI-EM8/IOVNET.PF0.VF0 io-1
primary# ldm add-io /SYS/RIO/NET2/IOVNET.PF0.VF0 io-1
```

4. I/O ドメインの IPMP グループに Ethernet 仮想機能を構成します。

- a. I/O ドメイン上の新しく追加されたネットワークデバイス (net1 と net2) を特定します。

```
io-1# dladm show-phys
LINK      MEDIA      STATE    SPPED    DUPLEX    DEVICE
net0      Ethernet   up       0         unknown   vnet0
net1      Ethernet   up       1000     full      igbvf0
net2      Ethernet   up       1000     full      igbvf1
```

- b. 新しく追加されたネットワークデバイスに対する IP インタフェースを作成します。

```
io-1# ipadm create-ip net1
io-1# ipadm create-ip net2
```

- c. この 2 つのネットワークインタフェース用に ipmp0 IPMP グループを作成します。

```
io-1# ipadm create-ipmp -i net1 -i net2 ipmp0
```

- d. IPMP グループに IP アドレスを割り当てます。

この例では DHCP オプションを構成します。

```
io-1# ipadm create-addr -T dhcp ipmp0/v4
```

- e. IPMP グループインタフェースのステータスを確認します。

```
io-1# ipmpstat -g
```

例 7-28 MPxIO を使用したファイバチャネル SR-IOV 機能によるマルチパス構成

この例は、MPxIO を使用して、回復性 I/O ドメインに対応するファイバチャネル仮想機能デバイスを構成する方法を示しています。詳細は、[Managing SAN Devices and Multipathing Oracle Solaris 11.2](#)を参照してください。

- 異なるルートドメインに割り当てられている 2 つのファイバチャネル SR-IOV 物理機能を特定します。

この例では、root-1 および root-2 ルートドメインにファイバチャネル SR-IOV 物理機能があります。

```
primary# ldm list-io | grep root-1 | grep PF
/SYS/PCI-EM4/IOVFC.PF0          PF      pci_1      root-1
primary# ldm list-io | grep root-2 | grep PF
/SYS/PCI-EM15/IOVFC.PF0        PF      pci_2      root-2
```

- 2 つの仮想機能を、指定した物理機能のそれぞれに作成します。

詳細は、[125 ページの「ファイバチャネル SR-IOV 仮想機能を作成する方法」](#)を参照してください。

```
primary# ldm create-vf port-wwn=10:00:00:14:4f:fc:60:00 \
node-wwn=20:00:00:14:4f:fc:60:00 /SYS/PCI-EM4/IOVFC.PF0
Created new vf: /SYS/PCI-EM4/IOVFC.PF0.VF0
primary# ldm create-vf port-wwn=10:00:00:14:4f:fc:70:00 \
node-wwn=20:00:00:14:4f:fc:70:00 /SYS/PCI-EM15/IOVFC.PF0
Created new vf: /SYS/PCI-EM15/IOVFC.PF0.VF0
```

- 新しく作成した仮想機能を io-1 I/O ドメインに追加します。

```
primary# ldm add-io /SYS/PCI-EM4/IOVFC.PF0.VF0 io-1
primary# ldm add-io /SYS/PCI-EM15/IOVFC.PF0.VF0 io-1
```

4. `prtconf -v` コマンドを使用して、MPxIO が I/O ドメインで有効になっているかどうか確認します。

fp デバイスの出力に次のデバイスプロパティ設定が含まれている場合は、MPxIO が有効になっています。

```
mpxio-disable="no"
```

mpxio-disable プロパティが yes に設定されている場合は、`/etc/driver/drv/fp.conf` ファイルでこのプロパティの値を no に更新して、I/O ドメインをリブートします。

mpxio-disable デバイスプロパティが `prtconf -v` の出力に表示されない場合は、`/etc/driver/drv/fp.conf` ファイルに `mpxio-disable="no"` エントリを追加して、I/O ドメインをリブートします。

5. MPxIO グループのステータスを確認します。

```
io-l# mpathadm show LU
```

```
Logical Unit: /dev/rdisk/c0t600A0B80002A384600003D6B544EECD0d0s2
```

```
mpath-support: libmpscsi_vhci.so
```

```
Vendor: SUN
```

```
Product: CSM200_R
```

```
Revision: 0660
```

```
Name Type: unknown type
```

```
Name: 600a0b80002a384600003d6b544eecd0
```

```
Asymmetric: yes
```

```
Current Load Balance: round-robin
```

```
Logical Unit Group ID: NA
```

```
Auto Failback: on
```

```
Auto Probing: NA
```

```
Paths:
```

```
Initiator Port Name: 100000144ffc6000
```

```
Target Port Name: 201700a0b82a3846
```

```
Override Path: NA
```

```
Path State: OK
```

```
Disabled: no
```

```
Initiator Port Name: 100000144ffc7000
```

```
Target Port Name: 201700a0b82a3846
Override Path: NA
Path State: OK
Disabled: no
```

Target Port Groups:

```
ID: 1
Explicit Failover: yes
Access State: active
```

Target Ports:

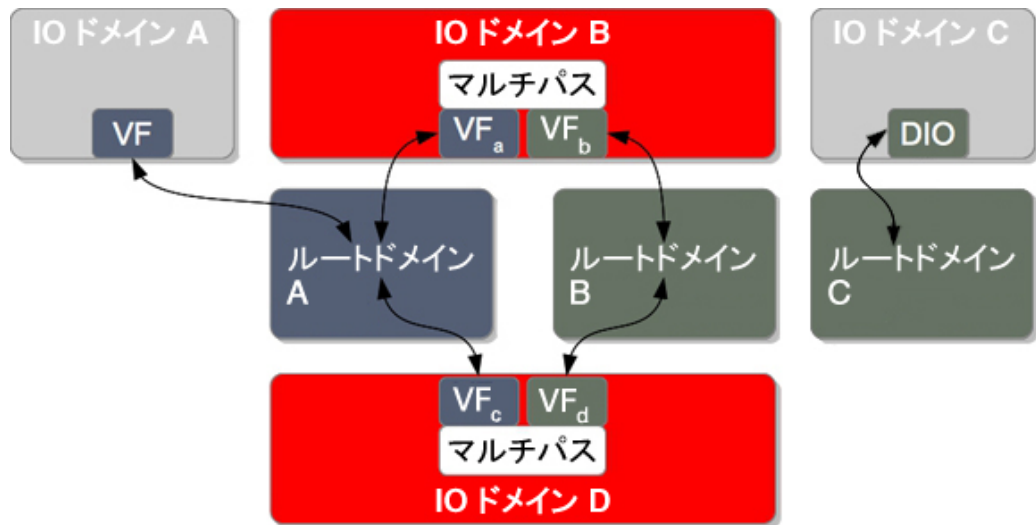
```
Name: 201700a0b82a3846
Relative ID: 0
```

例 – 回復性および非回復性構成の使用

回復性ドメインと非回復性ドメインの両方による構成を使用できます。

次の図は、I/O ドメイン A と I/O ドメイン C がどちらもマルチパスを使用していないために回復性を持たないことを示しています。I/O ドメイン A には仮想機能が含まれ、I/O ドメイン C には直接 I/O デバイスが含まれています。

図 7-2 回復性および非回復性 I/O ドメインによる構成



I/O ドメイン B および I/O ドメイン D には回復性があります。I/O ドメイン A、B、および D はルートドメイン A に依存します。I/O ドメイン B および D はルートドメイン B に依存します。I/O ドメイン C はルートドメイン C に依存します。

ルートドメイン A が中断されると、I/O ドメイン A も中断されます。ただし、I/O ドメイン B および D は代替パスにフェイルオーバーし、アプリケーションを実行し続けます。ルートドメイン C が中断された場合、I/O ドメイン C はルートドメイン C の `failure-policy` プロパティの値で指定された方法で失敗します。

非信頼性 I/O ドメインが構成された状態のルートドメインのリポート

注記 - I/O ドメインに回復性がある場合は、それをサービスするルートドメインが中断された場合でも、操作を続行できます。回復性 I/O ドメインの構成については、[136 ページの「I/O ドメインの回復性」](#)を参照してください。

I/O ドメイン内の PCIe スロットと同様に、[154 ページの「PCIe エンドポイントを構成した状態のルートドメインのリポート」](#)で説明している懸念事項は I/O ドメインに割り当てられている仮想機能にも関連します。

注記 - I/O ドメインは、関連するルートドメインが実行中でない場合は起動できません。

直接 I/O を使用した I/O ドメインの作成

この章では、直接 I/O に関する次のトピックについて説明します。

- 147 ページの「Creating an I/O Domain by Assigning PCIe Endpoint Devices」
- 150 ページの「直接 I/O のハードウェア要件とソフトウェア要件」
- 151 ページの「直接 I/O 機能の現在の制限事項」
- 152 ページの「PCIe エンドポイントデバイス構成の計画」
- 154 ページの「PCIe エンドポイントを構成した状態のルートドメインのリブート」
- 156 ページの「PCIe ハードウェアの変更」
- 159 ページの「PCIe エンドポイントデバイスを割り当てることによる I/O ドメインの作成」

Creating an I/O Domain by Assigning PCIe Endpoint Devices

個々の PCIe エンドポイントデバイス (または直接 I/O を割り当てが可能なデバイス) をドメインに割り当てることができます。PCIe エンドポイントデバイスをこのように使用することで、I/O ドメインにデバイスをより細かく割り当てることが可能になります。この機能は、直接 I/O (Direct I/O、DIO) 機能により提供されます。

DIO 機能によって、システム内の PCIe バスの数よりも多くの I/O ドメインを作成できます。I/O ドメインの最大数は現在、PCIe エンドポイントデバイス数によってのみ制限されます。

PCIe エンドポイントデバイスは、次のいずれかです。

- スロットの PCIe カード
- プラットフォームにより識別されるオンボードの PCIe デバイス

注記 - ルートドメインはその他のルートドメインとの依存関係を持つことができないため、PCIe バスを所有するルートドメインは、その PCIe エンドポイントデバイスまたは SR-IOV 仮想機能をほかのルートドメインに割り当てることができません。ただし、PCIe バスから、そのバスを所有するルートドメインに、PCIe エンドポイントデバイスまたは仮想機能を割り当てることができません。

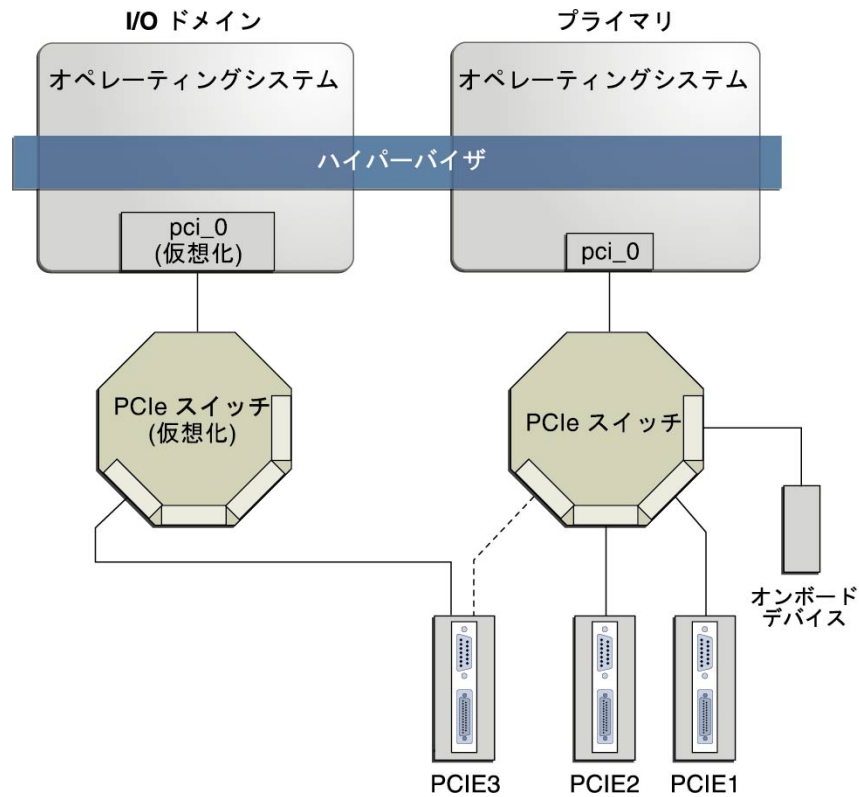
次の図は、PCIe エンドポイントデバイス PCIE3 が I/O ドメインに割り当てられている状態を示しています。I/O ドメインのバス pci_0 とスイッチは、いずれも仮想のものです。PCIE3 エンドポイントデバイスには、primary ドメインではアクセスできなくなっています。

I/O ドメインで、pci_0 ブロックとスイッチは、それぞれ仮想ルートコンプレックスと仮想 PCIe スイッチです。このブロックとスイッチは、primary ドメインの pci_0 ブロックとスイッチに似ています。primary ドメインで、スロット PCIE3 のデバイスは元のデバイスの「シャドウ」であり、SUNW, assigned として識別されます。



注意 - ldm remove-io コマンドを使用して PCIe エンドポイントデバイスを primary ドメインから削除したあと、Oracle Solaris ホットプラグ操作を使用してそのデバイスをホットリムーブすることはできません。PCIe エンドポイントデバイスの交換または取り外しの詳細は、[156 ページの「PCIe ハードウェアの変更」](#)を参照してください。

図 8-1 PCIe エンドポイントデバイスの I/O ドメインへの割り当て



ldm list-io コマンドを使用して、PCIe エンドポイントデバイスの一覧を確認します。

DIO 機能により、スロットの任意の PCIe カードを I/O ドメインに割り当てることができますが、サポートされるのは特定の PCIe カードのみです。[150 ページの「直接 I/O のハードウェア要件とソフトウェア要件」](#)を参照してください。



注意 - ブリッジを持つ PCIe カードはサポートされません。PCIe の機能レベルの割り当てでもサポートされません。サポートされていない PCIe カードを I/O ドメインに割り当てた場合、予期しない動作が引き起こされることがあります。

次の項目は、DIO 機能についての重要な詳細です。

- この機能は、ソフトウェアのすべての要件が満たされている場合にのみ有効です。[150 ページの「直接 I/O のハードウェア要件とソフトウェア要件」](#)を参照してください。
- ルートドメインに割り当てられた PCIe バスに接続されている PCIe エンドポイントのみが、DIO 機能によってほかのドメインに割り当て可能です。
- DIO を使用している I/O ドメインは、ルートドメインの実行中のみ PCIe エンドポイントデバイスにアクセスできます。
- ルートドメインをリブートすると、PCIe エンドポイントデバイスが存在する I/O ドメインに影響が及びます。[154 ページの「PCIe エンドポイントを構成した状態のルートドメインのリブート」](#)を参照してください。ルートドメインは、次のタスクも実行します。
 - PCIe バスを初期化して管理します。
 - I/O ドメインに割り当てられている PCIe エンドポイントデバイスにより引き起こされたすべてのバスエラーを処理する。PCIe バスに関連するすべてのエラーを受け取るのは primary ドメインのみであることに注意してください。

直接 I/O のハードウェア要件とソフトウェア要件

直接 I/O (DIO) 機能を使用してドメインに直接 I/O デバイスを割り当てるには、適切なソフトウェアを実行することと、サポートされている PCIe カードを使用することが必要です。

- **ハードウェア要件。**I/O ドメイン上で直接 I/O エンドポイントデバイスとして使用できるのは、特定の PCIe カードのみです。それ以外のカードを Oracle VM Server for SPARC 環境で使用することもできますが、DIO 機能とともに使用できません。代わりに、サービスドメインに使用することや、ルートコンプレックス全体が割り当てられた I/O ドメインに使用することは可能です。

使用しているプラットフォームで利用できるカードを確認するには、プラットフォームのハードウェアドキュメントを参照してください。サポートされる PCIe カードの最新のリストについては、<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1> を参照してください。

注記 - SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーには複数の PCIe バスを提供する I/O コントローラがあるため、PCIe バスの割り当て機能を使用して PCIe カードを異なるドメインに割り当てることができます。詳細は、[第6章「PCIe バスの割り当てによってルートドメインを作成する方法」](#)を参照してください。

- **ソフトウェア要件。**DIO 機能を使用するには、次のドメインで、サポートされる OS が実行されている必要があります。

- **ルートドメイン。**Oracle Solaris 11.3 OS 以上。

推奨される方法として、すべてのドメインで少なくとも Oracle Solaris 10 1/13 OS に加えて「[Fully Qualified Oracle Solaris OS Versions](#)」 in 『[Oracle VM Server for SPARC 3.3 Installation Guide](#)』にある必須パッチ、または Oracle Solaris 11.3 OS を実行します。

- **I/O ドメイン。**Oracle Solaris 11 OS 以上。最新の Oracle Solaris 11 リリースには追加機能のサポートが含まれていることに注意してください。

注記 - プラットフォームでサポートされているすべての PCIe カードは、ルートドメインでサポートされます。サポートされている PCIe カードの一覧は、お使いのプラットフォームのドキュメントを参照してください。ただし、I/O ドメインに割り当てることができるのは、直接 I/O がサポートされている PCIe カードのみです。

直接 I/O 機能を使用して PCIe エンドポイントデバイスを追加または削除するには、まず PCIe バス自体で I/O 仮想化を有効にする必要があります。

ldm set-io または ldm add-io コマンドを使用すると、iov プロパティを on に設定できます。また、ldm add-domain または ldm set-domain コマンドを使用すると、rc-add-policy プロパティを iov に設定できます。[ldm\(1M\)](#) マニュアルページを参照してください。

ルートドメインをリブートすると直接 I/O に影響を及ぼすため、直接 I/O の構成変更は、直接 I/O に関連するルートドメインの変更が最大限になり、ルートドメインのリブートが最小限になるように、十分に計画してください。

直接 I/O 機能の現在の制限事項

制限を回避する方法については、[152 ページの「PCIe エンドポイントデバイス構成の計画」](#)を参照してください。

ルート以外のドメインに PCIe エンドポイントデバイスを割り当てるか削除する操作は、そのドメインが停止中または非アクティブの場合のみ許可されます。

注記 - Fujitsu M10 サーバー は、PCIe エンドポイントデバイスの動的再構成をサポートします。ルートドメインをリブートしたり I/O ドメインを停止したりせずに、PCIe エンドポイントデバイスを割り当てまたは削除できます。

この機能の最新情報は、お使いのモデルに関する *Fujitsu M10/SPARC M10 Systems のシステム運用と管理ガイド* (<http://www.fujitsu.com/global/services/computing/server/sparc/downloads/manual/>) を参照してください。

注記 - 直接 I/O 機能は、SPARC M7 シリーズサーバーおよび SPARC T7 シリーズサーバーではサポートされていません。代わりに、PCIe バスの割り当て機能を使用してください。[第6章「PCIe バスの割り当てによってルートドメインを作成する方法」](#)を参照してください。

SPARC T5 および SPARC M6 プラットフォームまでの SPARC システムでは、提供される割り込み数が制限されているので、Oracle Solaris は各デバイスが使用可能な割り込み数を制限します。デフォルトの制限は一般的なシステム構成のニーズに一致する必要がありますが、特定のシステム構成ではこの値の調整が必要になることがあります。[421 ページの「割り込み制限の調整」](#)を参照してください。

PCIe エンドポイントデバイス構成の計画

ルートドメインの停止を避けるため、PCIe エンドポイントデバイスの割り当てまたは削除は、事前に慎重に計画します。ルートドメインのリブートはルートドメイン自体で利用可能なサービスに影響するだけでなく、PCIe エンドポイントデバイスを割り当てる I/O ドメインにも影響します。個々の I/O ドメインへの変更は、他のドメインに影響を及ぼしませんが、事前に計画することにより、そのドメインによって提供されるサービスへの影響を最小限に抑えることができます。

遅延再構成では、デバイスの追加または削除を引き続き行い、そのあとでルートドメインを 1 回だけリブートすることで、すべての変更を有効にできます。

例については、[159 ページの「PCIe エンドポイントデバイスを割り当てることによって I/O ドメインを作成する方法」](#)を参照してください。

DIO デバイス構成の計画と実行には、一般に次のような手順に従う必要があります。

1. システムのハードウェア構成を理解し、記録します。

具体的には、システムの PCIe カードについて、部品番号やその他の詳細情報を記録します。

`ldm list-io -l` および `prtdiag -v` コマンドを使用して情報を取得し、あとで参照するために保存します。

2. primary ドメインに必要な PCIe エンドポイントデバイスを特定します。

たとえば、次のデバイスへのアクセスを提供する PCIe エンドポイントデバイスを特定します。

- ブートディスクデバイス
- ネットワークデバイス
- primary ドメインがサービスとして提供するその他のデバイス

3. I/O ドメインで使用する可能性のあるすべての PCIe エンドポイントデバイスを削除します。

この手順により、以後ルートドメインでリブート操作を実行することを避け、リブートによる I/O ドメインへの影響を防ぐことができます。

PCIe エンドポイントデバイスを削除するには、`ldm remove-io` コマンドを使用します。`remove-io` および `add-io` サブコマンドでデバイスを指定するには、デバイスパスの代わりに仮名を使用します。

注記 - 遅延再構成時に、必要なすべてのデバイスを削除したあとですべての変更を有効にするには、ルートドメインを 1 回だけリブートする必要があります。

4. この構成をサービスプロセッサ (SP) に保存します。

`ldm add-config` コマンドを使用します。

5. ルートドメインをリブートし、段階 3 で削除した PCIe エンドポイントデバイスを解放します。
6. 削除した PCIe エンドポイントデバイスが、ルートドメインに割り当てられていないことを確認します。

`ldm list-io -l` コマンドを使用して、削除したデバイスが出力に `SUNW,assigned-device` として表示されることを確認します。

7. 物理デバイスへの直接アクセスを可能にするため、使用可能な PCIe エンドポイントデバイスをゲストドメインに割り当てます。

この割り当てを行なったあとで、ドメインの移行機能を使用して別の物理システムにゲストドメインを移行することはできません。

8. ゲストドメインに PCIe エンドポイントデバイスを追加するか、ゲストドメインから PCIe エンドポイントデバイスを削除します。

`ldm add-io` コマンドを使用します。

I/O ドメインへの変更を最小限に抑えることで、リブート操作を減らし、そのドメインが提供するサービスの停止を避けます。

9. (オプション) PCIe ハードウェアに変更を加えます。

[156 ページの「PCIe ハードウェアの変更」](#)を参照してください。

PCIe エンドポイントを構成した状態のルートドメインのリブート

ルートドメインは PCIe バスの所有者で、バスの初期化と管理の役割を担います。ルートドメインはアクティブで、DIO または SR-IOV 機能をサポートするバージョンの Oracle Solaris OS を実行している必要があります。ルートドメインのシャットダウン、停止、またはリブートを行うと、PCIe バスへのアクセスが中断されます。PCIe バスが使用できないとき、そのバス上の PCIe デバイスが影響を受け、使用不可能になることがあります。

PCIe エンドポイントデバイスを持つ I/O ドメインの実行中にルートドメインがリブートされた場合、I/O ドメインの動作は予測不能です。たとえば、リブート中またはリブート後に、PCIe エンドポイントデバイスを持つ I/O ドメインでパニックが発生することがあります。ルートドメインをリブートするときは、各ドメインを手動で停止/開始する必要があります。

I/O ドメインに回復性があれば、PCIe バスの所有者であるルートドメインが使用できなくなった場合でも動作を続行できることに注意してください。[136 ページの「I/O ドメインの回復性」](#)を参照してください。

注記 - I/O ドメインは、関連するルートドメインが実行中でない場合は起動できません。

これらの問題を回避するには、次のいずれかの手順を実行します。

- ルートドメインをシャットダウンする前に、システムの PCIe エンドポイントデバイスが割り当てられているドメインすべてを手動でシャットダウンします。

この手順により、ルートドメインのシャットダウン、停止、またはリブートを行う前に、それらのドメインを確実にクリーンにシャットダウンします。

PCIe エンドポイントデバイスが割り当てられているすべてのドメインを調べるには、`ldm list-io` コマンドを実行します。このコマンドにより、システムのドメインに割り当てられてい

る PCIe エンドポイントデバイスを一覧表示できます。このコマンドによる出力の詳細な説明については、[ldm\(1M\)](#) マニュアルページを参照してください。

見つかったそれぞれのドメインについて、`ldm stop` コマンドを実行してドメインを停止します。

- ルートドメインと、PCIe エンドポイントデバイスが割り当てられているドメインの間の、ドメインの依存関係を構成します。

この依存関係により、何かの理由でルートドメインがリブートしたときに、PCIe エンドポイントデバイスを持つドメインが確実に自動的にリブートされます。

それらのドメインはこの依存関係によって強制的にリセットされるため、クリーンなシャットダウンはできません。ただし、依存関係は、手動でシャットダウンされたドメインには影響を及ぼしません。

```
primary# ldm set-domain failure-policy=reset primary
```

```
primary# ldm set-domain master=primary domain-name
```

例 8-1 `primary` 以外のルートドメインおよび I/O ドメインによる構成の障害ポリシーの依存関係の構成

次の例では、`primary` 以外のルートドメインおよび I/O ドメインを使用する構成で、障害ポリシーの依存関係を構成できる方法について説明します。

この例で、`ldg1` は `primary` 以外のルートドメイン、`ldg2` は `ldg1` ドメインが所有するルートコンプレックスから割り当てられた PCIe SR-IOV 仮想機能または PCIe エンドポイントデバイスを持つ I/O ドメインです。

```
primary# ldm set-domain failure-policy=stop ldg1
```

```
primary# ldm set-domain master=ldg1 ldg2
```

この依存関係により、`ldg1` ルートドメインをリブートするときには I/O ドメインが停止します。

- `primary` 以外のルートドメインがリブートする場合、この依存関係により I/O ドメインが停止します。`primary` 以外のルートドメインがブートするときに、I/O ドメインが起動します。

```
primary# ldm start ldg2
```

- `primary` ルートドメインをリブートする場合は、このポリシー設定により `primary` 以外のルートドメインおよび従属する I/O ドメインの両方が停止します。`primary` ドメインがブートするときは、`primary` 以外のルートドメインを先に起動する必要があります。ドメインがブートするときに、I/O ドメインを起動します。

```
primary# ldm start ldg1
```

ldg1 ドメインがアクティブになるのを待ってから、I/O ドメインを起動します。

```
primary# ldm start ldg2
```

PCIe ハードウェアの変更

以下の手順は、PCIe エンドポイント割り当ての構成の間違いを防ぐために役立ちます。特定のハードウェアのインストールと削除に関するプラットフォーム固有の情報については、お使いのプラットフォームのドキュメントを参照してください。

- 空のスロットに PCIe カードを取り付ける場合は、作業は必要ありません。この PCIe カードは自動的に、PCIe バスを所有するドメインにより所有されます。

I/O ドメインに新しい PCIe カードを割り当てるには、`ldm remove-io` コマンドを使用して、まずルートドメインからカードを削除します。そのあとで、`ldm add-io` コマンドを使用してカードを I/O ドメインに割り当てます。

- PCIe カードがシステムから削除され、ルートドメインに割り当てられるときは、アクションは必要ありません。
- I/O ドメインに割り当てられている PCIe カードを削除するには、最初に I/O ドメインからデバイスを削除します。そのあとで、システムからデバイスを物理的に取り除く前に、そのデバイスをルートドメインに追加します。
- I/O ドメインに割り当てられている PCIe カードを置き換えるには、新しいカードが DIO 機能でサポートされていることを確認します。

サポートされている場合は、現在の I/O ドメインに新しいカードが自動的に割り当てられ、作業は必要ありません。

サポートされていない場合は、`ldm remove-io` コマンドを使用して、最初にその PCIe カードを I/O ドメインから削除します。次に `ldm add-io` コマンドを使用して、その PCIe カードをルートドメインに再割り当てします。そのあとで、ルートドメインに割り当てた PCIe カードを別の PCIe カードに物理的に置き換えます。これらの手順により、DIO 機能によってサポートされていない構成を回避することができます。

PCIe カードを削除するときのゲストドメインの停止の最小化

Oracle VM Server for SPARC ソフトウェアを実行するシステムから PCIe カードを削除または交換するときに、このハードウェアに依存するドメインは使用できません。ゲストドメインの停止を最小限に抑えるには、カードを物理的に削除するためにはホットプラグ機能を使用するようにシステムを準備する必要があります。

▼ PCIe カードを削除するときのゲストドメインの停止を最小限にする方法

この手順により、直接 I/O または SR-IOV デバイスが割り当てられておらず、複数のパスが構成されているゲストドメインの停止を避けることができます。この手順では、primary ドメインを 2 回リブートする必要があります。

注記 - この手順は、PCIe カードが primary ルートドメイン以外の所有するルートコンプレックス上にある場合は当てはまりません。この場合は、代わりに [How to Replace PCIe Direct I/O Cards Assigned to an Oracle VM Server for SPARC Guest Domain \(Doc ID 1684273.1\)](https://support.oracle.com/epmos/faces/DocumentDisplay?_afLoop=226878266536565&id=1684273.1&_adf.ctrl-state=bo9fbmr1n_49) (https://support.oracle.com/epmos/faces/DocumentDisplay?_afLoop=226878266536565&id=1684273.1&_adf.ctrl-state=bo9fbmr1n_49) を参照してください。

1. PCIe スロットが割り当てられているゲストドメインを停止します。

```
primary# ldm stop domain-name
```

2. ゲストドメインから PCIe スロットを削除します。

```
primary# ldm remove-io PCIe-slot domain-name
```

3. PCIe スロットと SR-IOV 仮想機能が割り当てられているゲストドメインを停止します。

```
primary# ldm stop domain-name
```

注記 - PCIe バスが割り当てられているゲストドメインは、ネットワークへの代替パスとゲストドメインへのディスクデバイスを提供している可能性があるため、停止する必要はありません。

4. primary ドメインで遅延再構成を開始して、このスロットをここに割り当てできるようにします。

```
primary# ldm start-reconf primary
```

5. primary ドメインに PCIe スロットを追加します。

```
primary# ldm add-io PCIe-slot domain-name
```

6. **primary** ドメインをリブートします。

```
primary# shutdown -i6 -g0 -y
```

7. ホットプラグのコマンドを使用して、PCIe カードを交換します。

Oracle Solaris OS のホットプラグ機能の詳細は、『[Managing Devices in Oracle Solaris 11.3](#)』の第 2 章「[Dynamically Configuring Devices](#)」を参照してください。

8. カードの交換後、この同じ PCIe スロットをゲストドメインに再割り当てする必要がある場合は、次の手順を実行します。

- a. **primary** ドメインの遅延再構成を開始します。

```
primary# ldm start-reconf primary
```

- b. **primary** ドメインから PCIe スロットを削除します。

```
primary# ldm remove-io PCIe-slot domain-name
```

- c. PCIe スロットの削除を有効にするために、**primary** ドメインをリブートします。

```
primary# shutdown -i6 -g0 -y
```

- d. PCIe スロットをゲストドメインに再割り当てします。

```
primary# ldm add-io PCIe-slot domain-name
```

- e. PCIe スロットと SR-IOV 仮想機能を割り当てるゲストドメインを起動します。

```
primary# ldm start domain-name
```

PCIe エンドポイントデバイスを割り当てることによる I/O ドメインの作成

▼ PCIe エンドポイントデバイスを割り当てることによって I/O ドメインを作成する方法

停止時間を最小限に抑えるため、DIO の導入はすべて事前に計画します。



注意 - SPARC T3-1 または SPARC T4-1 システム上の /SYS/MB/SASHBA1 スロットを DIO ドメインに割り当てると、primary ドメインはオンボードの DVD デバイスへのアクセスを失います。

SPARC T3-1 および SPARC T4-1 システムには、オンボードのストレージ用に 2 基の DIO スロットが搭載されており、それらは /SYS/MB/SASHBA0 および /SYS/MB/SASHBA1 パスで表されます。/SYS/MB/SASHBA1 スロットは、複数ヘッドのオンボードのディスクに加え、オンボードの DVD デバイスも収容します。そのため、/SYS/MB/SASHBA1 を DIO ドメインに割り当てると、primary ドメインはオンボードの DVD デバイスへのアクセスを失います。

SPARC T3-2 および SPARC T4-2 システムには 1 つの SASHBA スロットがあり、それにすべてのオンボードのディスクとオンボードの DVD デバイスが収容されます。そのため、SASHBA を DIO ドメインに割り当てると、オンボードのディスクおよびオンボードの DVD デバイスが DIO ドメインに貸し出されるため、primary ドメインには利用できなくなります。

PCIe エンドポイントデバイスを追加して I/O ドメインを作成する例については、[152 ページの「PCIe エンドポイントデバイス構成の計画」](#)を参照してください。

注記 - 今回のリリースでは、DefaultFixed NCP を使用して、Oracle Solaris 11 システム上にデータリンクおよびネットワークインタフェースを構成します。

Oracle Solaris 11 OSには、次の NCP が含まれています。

- DefaultFixed – dladm または ipadm コマンドを使用して、ネットワークを管理できます
- Automatic – netcfg または netadm コマンドを使用して、ネットワークを管理できます

netadm list コマンドを使用して、DefaultFixed NCP が有効になっていることを確認します。『[Oracle Solaris Administration: Network Interfaces and Network Virtualization](#)』の第 7 章「Using Datalink and Interface Configuration Commands on Profiles」を参照してください。

1. システムに現在インストールされているデバイスを特定し、保存します。

ldm list-io -l コマンドの出力に、現在の I/O デバイスの構成が表示されます。prtdiag -v コマンドを使用すると、より詳細な情報を得ることができます。

注記 - I/O ドメインにデバイスが割り当てられたあとでは、デバイスの ID は I/O ドメインでのみ特定できます。

```
primary# ldm list-io -l
```

NAME	TYPE	BUS	DOMAIN	STATUS
niu_0	NIU	niu_0	primary	
[niu@480]				
niu_1	NIU	niu_1	primary	
[niu@580]				
pci_0	BUS	pci_0	primary	
[pci@400]				
pci_1	BUS	pci_1	primary	
[pci@500]				
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
[pci@400/pci@2/pci@0/pci@8]				
SUNW,emlxs@0/fp/disk				
SUNW,emlxs@0/fp/tape				
SUNW,emlxs@0/fp@0,0				
SUNW,emlxs@0,1/fp/disk				
SUNW,emlxs@0,1/fp/tape				
SUNW,emlxs@0,1/fp@0,0				
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
[pci@400/pci@2/pci@0/pci@4]				
pci/scsi/disk				
pci/scsi/tape				
pci/scsi/disk				
pci/scsi/tape				
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC


```

[pci@400/pci@2/pci@0/pci@0]
    ethernet@0
    ethernet@0,1
    SUNW,qlc@0,2/fp/disk
    SUNW,qlc@0,2/fp@0,0
    SUNW,qlc@0,3/fp/disk
    SUNW,qlc@0,3/fp@0,0
/SYS/MB/PCIE6                                PCIE   pci_0   primary EMP
[pci@400/pci@1/pci@0/pci@8]
/SYS/MB/PCIE8                                PCIE   pci_0   primary EMP
[pci@400/pci@1/pci@0/pci@c]
/SYS/MB/SASHBA                                PCIE   pci_0   primary OCC
[pci@400/pci@2/pci@0/pci@e]
    scsi@0/iport@1
    scsi@0/iport@2
    scsi@0/iport@4
    scsi@0/iport@8
    scsi@0/iport@80/cdrom@p7,0
    scsi@0/iport@v0
/SYS/MB/NET0                                PCIE   pci_0   primary OCC
[pci@400/pci@1/pci@0/pci@4]
    network@0
    network@0,1
/SYS/MB/PCIE1                                PCIE   pci_1   primary OCC
[pci@500/pci@2/pci@0/pci@a]
    SUNW,qlc@0/fp/disk
    SUNW,qlc@0/fp@0,0
    SUNW,qlc@0,1/fp/disk
    SUNW,qlc@0,1/fp@0,0
/SYS/MB/PCIE3                                PCIE   pci_1   primary OCC
[pci@500/pci@2/pci@0/pci@6]
    network@0
    network@0,1
    network@0,2
    network@0,3
/SYS/MB/PCIE5                                PCIE   pci_1   primary OCC
[pci@500/pci@2/pci@0/pci@0]
    network@0
    network@0,1
/SYS/MB/PCIE7                                PCIE   pci_1   primary EMP
[pci@500/pci@1/pci@0/pci@6]
/SYS/MB/PCIE9                                PCIE   pci_1   primary EMP
[pci@500/pci@1/pci@0/pci@0]
/SYS/MB/NET2                                PCIE   pci_1   primary OCC
[pci@500/pci@1/pci@0/pci@5]
    network@0
    network@0,1
    ethernet@0,80
/SYS/MB/NET0/IOVNET.PF0                      PF      pci_0   primary
[pci@400/pci@1/pci@0/pci@4/network@0]
    maxvfs = 7
/SYS/MB/NET0/IOVNET.PF1                      PF      pci_0   primary
[pci@400/pci@1/pci@0/pci@4/network@0,1]
    maxvfs = 7

```

```

/SYS/MB/PCIE5/IOVNET.PF0          PF    pci_1    primary
[pci@500/pci@2/pci@0/pci@0/network@0]
    maxvfs = 63
/SYS/MB/PCIE5/IOVNET.PF1          PF    pci_1    primary
[pci@500/pci@2/pci@0/pci@0/network@0,1]
    maxvfs = 63
/SYS/MB/NET2/IOVNET.PF0          PF    pci_1    primary
[pci@500/pci@1/pci@0/pci@5/network@0]
    maxvfs = 7
/SYS/MB/NET2/IOVNET.PF1          PF    pci_1    primary
[pci@500/pci@1/pci@0/pci@5/network@0,1]
    maxvfs = 7

```

2. 保持する必要があるブートディスクのデバイスパスを確認します。

66 ページの「PCIe バスの割り当てによってルートドメインを作成する方法」の手順 2 を参照してください。

3. ブロック型デバイスが接続されている物理デバイスを確認します。

66 ページの「PCIe バスの割り当てによってルートドメインを作成する方法」の手順 3 を参照してください。

4. システムで使用されているネットワークインタフェースを確認します。

66 ページの「PCIe バスの割り当てによってルートドメインを作成する方法」の手順 4 を参照してください。

5. ネットワークインタフェースが接続されている物理デバイスを確認します。

次のコマンドでは、igb0 ネットワークインタフェースを使用します。

```

primary# ls -l /dev/igb0
lrwxrwxrwx  1 root    root          46 Jul 30 17:29 /dev/igb0 ->
../devices/pci@500/pci@0/pci@8/network@0:igb0

```

この例で、primary ドメインによって使用されるネットワークインタフェースの物理デバイスは、手順 1 の MB/NET0 一覧に対応する PCIe エンドポイントデバイス (pci@500/pci@0/pci@8) に接続されています。そのため、このデバイスを primary ドメインから削除しないでください。他の PCIe デバイスはすべて、primary ドメインで使用されないため、他のドメインに安全に割り当てることができます。

primary ドメインで使用されているネットワークインタフェースが、別のドメインに割り当てようとしているバス上にある場合は、別のネットワークインタフェースを使用するように primary ドメインを再構成する必要があります。

6. I/O ドメインで使用する可能性がある PCIe エンドポイントデバイスを削除します。

この例では、PCIE2、PCIE3、PCIE4、および PCIE5 エンドポイントデバイスが primary ドメインで使用されないため、これらを削除できます。

a. PCIe エンドポイントデバイスを削除します。



注意 - primary ドメインで使用されるデバイス、または必要なデバイスは、削除しないでください。ドメインによって使用されているデバイス (ネットワークポートや usbecm デバイスなど) を含むバスを削除しないでください。

誤ってデバイスを削除してしまった場合は、`ldm cancel-reconf primary` コマンドを使用して、primary ドメインでの遅延再構成を取り消します。

リブートの繰り返しを避けるために、同時に複数のデバイスを削除できます。

```
primary# ldm start-reconf primary
primary# ldm set-io iov=on pci_1
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
primary# ldm remove-io /SYS/MB/PCIE1 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
primary# ldm remove-io /SYS/MB/PCIE3 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
primary# ldm remove-io /SYS/MB/PCIE5 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

b. 新しい構成をサービスプロセッサ (SP) に保存します。

次のコマンドにより、dio という名前のファイルに構成が保存されます。

```
primary# ldm add-config dio
```

c. システムをリブートし、PCIe エンドポイントデバイスの削除を反映させます。

```
primary# shutdown -i6 -g0 -y
```

7. primary ドメインにログインし、PCIe エンドポイントデバイスのドメインへの割り当てが解除されたことを確認します。

```
primary# ldm list-io
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
niu_0                              NIU   niu_0   primary
niu_1                              NIU   niu_1   primary
pci_0                              BUS   pci_0   primary
pci_1                              BUS   pci_1   primary  IOV
/SYS/MB/PCIE0                      PCIE  pci_0   primary  OCC
/SYS/MB/PCIE2                      PCIE  pci_0   primary  OCC
/SYS/MB/PCIE4                      PCIE  pci_0   primary  OCC
/SYS/MB/PCIE6                      PCIE  pci_0   primary  EMP
/SYS/MB/PCIE8                      PCIE  pci_0   primary  EMP
/SYS/MB/SASHBA                     PCIE  pci_0   primary  OCC
/SYS/MB/NET0                       PCIE  pci_0   primary  OCC
/SYS/MB/PCIE1                      PCIE  pci_1   primary  OCC
/SYS/MB/PCIE3                      PCIE  pci_1   primary  OCC
/SYS/MB/PCIE5                      PCIE  pci_1   primary  OCC
/SYS/MB/PCIE7                      PCIE  pci_1   primary  EMP
/SYS/MB/PCIE9                      PCIE  pci_1   primary  EMP
/SYS/MB/NET2                       PCIE  pci_1   primary  OCC
/SYS/MB/NET0/IOVNET.PF0            PF    pci_0   primary
/SYS/MB/NET0/IOVNET.PF1            PF    pci_0   primary
/SYS/MB/NET2/IOVNET.PF0            PF    pci_1   primary
/SYS/MB/NET2/IOVNET.PF1            PF    pci_1   primary
```

注記 - ldm list-io -l の出力で、削除された PCIe エンドポイントデバイスについて SUNW, assigned-device と表示されることがあります。実際の情報は primary ドメインから取得できなくなりますが、デバイスが割り当てられたドメインにはこの情報が存在しています。

8. PCIe エンドポイントデバイスをドメインに割り当てます。

a. PCIE3 デバイスを ldg1 ドメインに追加します。

```
primary# ldm add-io /SYS/MB/PCIE3 ldg1
```

b. ldg1 ドメインをバインドし、起動します。

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
```

9. ldg1 ドメインにログインし、デバイスが使用可能であることを確認します。

ネットワークデバイスが使用可能であることを確認してから、ネットワークデバイスをドメインで使用するための構成を行います。

```
primary# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
net0          Ethernet      unknown  0      unknown nxge0
```

net1	Ethernet	unknown	0	unknown	nxge1
net2	Ethernet	unknown	0	unknown	nxge2
net3	Ethernet	unknown	0	unknown	nxge3

primary 以外のルートドメインの使用

この章では、次の primary 以外のルートドメインのトピックについて説明します。

- [167 ページの「primary 以外のルートドメインの概要」](#)
- [169 ページの「primary 以外のルートドメインの要件」](#)
- [169 ページの「primary 以外のルートドメインの制限」](#)
- [170 ページの「primary 以外のルートドメインの例」](#)

primary 以外のルートドメインの概要

ルートドメインには PCIe ルートコンプレックスが割り当てられます。このドメインは PCIe ファブリックを所有し、ファブリックのエラー処理など、ファブリックに関連するすべてのサービスを提供します。ルートドメインは I/O ドメインでもあり、物理 I/O デバイスを所有し、それらに直接アクセスできます。デフォルトのルートドメインは、primary ドメインです。

任意のルートドメインに割り当てられている PCIe バス上で、直接 I/O および SR-IOV 操作を実行できます。primary 以外のルートドメインを含むすべてのルートドメインに対して次の操作を実行できるようになりました。

- PCIe スロットのステータスを表示する
- 存在する SR-IOV 物理機能を表示する
- I/O ドメインまたはルートドメインに PCIe スロットを割り当てる
- I/O ドメインまたはルートドメインから PCIe スロットを削除する
- 物理機能から仮想機能を作成する
- 仮想機能を破棄する
- 別のドメインに仮想機能を割り当てる

■ 別のドメインから仮想機能を削除する

Logical Domains Manager は、primary 以外のルートドメインで実行する Logical Domains エージェントから PCIe エンドポイントデバイスおよび SR-IOV 物理機能デバイスを取得します。この情報は、ルートドメインが最初に検出されてからルートドメインがブートされるまでの停止している間にキャッシュされます。

直接 I/O および SR-IOV 操作は、ルートドメインがアクティブな場合にのみ実行できます。Logical Domains Manager はその時に存在している実際のデバイスでのみ動作します。キャッシュされたデータは、次の操作が行われたときにリフレッシュされることがあります。

- Logical Domains エージェントが、指定したルートドメインで再起動される
- ホットプラグ操作などのハードウェアの変更が、指定したルートドメインで実行される

ldm list-io コマンドを使用して、PCIe エンドポイントデバイスのステータスを表示します。この出力には、primary 以外の各ルートドメインによって所有されているルートコンプレックスからのサブデバイスおよび物理機能デバイスも示されます。

次のコマンドは、どのルートドメインにも適用できます。

- ldm add-io
- ldm remove-io
- ldm set-io
- ldm create-vf
- ldm destroy-vf
- ldm start-reconf
- ldm cancel-reconf

遅延再構成のサポートは、primary 以外のルートドメインを含むように拡張されました。ただし、それを使用できるのは、ldm add-io、ldm remove-io、ldm set-io、ldm create-vf、および ldm destroy-vf コマンドを実行する場合のみです。遅延再構成は、次のような動的操作を使用して、実行できないすべての操作に使用できます。

- 直接 I/O 操作の実行
- 動的 SR-IOV 構成の要件を満たしていない物理機能からの仮想機能の作成および破棄。



注意 - ルートドメインのリブート回数を最小限に抑えて、停止時間を最小限に抑えるように事前に計画してください。

primary 以外のルートドメインの要件

直接 I/O および SR-IOV 機能をほかのドメインに提供するために、制御ドメインに加えて primary 以外のルートドメインを使用できます。この機能は、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでサポートされています。

■ ハードウェア要件。

[https://support.oracle.com/CSP/main/article?](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1)

[cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1) に記載されている直接 I/O および SR-IOV 機能用の PCIe カードに加えて、ほかの PCIe カードも使用できますが、DIO および SR-IOV 機能用ではありません。使用しているプラットフォームで利用できるカードを調べるには、プラットフォームのドキュメントを参照してください。

■ ファームウェア要件。

SPARC T4 プラットフォームでは、少なくともバージョン 8.4.0.a のシステムファームウェアが実行されている必要があります。

SPARC T5 サーバー、SPARC M5 サーバー、および SPARC M6 サーバーは、少なくともバージョン 9.1.0.x のシステムファームウェアを実行している必要があります。

SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーは、少なくともバージョン 9.4.3 のシステムファームウェアを実行している必要があります。

Fujitsu M10 サーバーでは、少なくともバージョン XCP2210 のシステムファームウェアが実行されている必要があります。

■ ソフトウェア要件。

primary 以外のドメインは、少なくとも Oracle Solaris 11.2 OS を実行している必要があります。

primary 以外のルートドメインの制限

primary 以外のルートドメインの使用には次の制限があります。

- I/O ドメインは、関連するルートドメインが実行中でない場合は起動できません。
- 遅延再構成のサポートは、primary 以外のルートドメインに拡張されました。ルートドメインがリブートされるか、遅延再構成が取り消されるまで、次のコマンドしか実行できません。

- `ldm add-io`
- `ldm remove-io`
- `ldm set-io`
- `ldm create-vf`
- `ldm destroy-vf`
- 次の操作を実行するには、ルートドメインがアクティブで、ブートされている必要があります。
 - SR-IOV 仮想機能の作成および破棄
 - PCIe スロットの追加および削除
 - SR-IOV 仮想機能の追加および削除
- PCIe スロットの `ldm add-io` および `ldm remove-io` 直接 I/O 操作を実行する場合、ルートドメインで遅延再構成を開始する必要があります。
- 構成が動的 I/O 仮想化の要件を満たしていない場合は、次の SR-IOV 仮想機能の操作に、遅延再構成を使用する必要があります。
 - `ldm create-vf`
 - `ldm destroy-vf`
 - `ldm add-io`
 - `ldm remove-io`
 - `ldm set-io`
- ルートドメインのリブートは、ルートドメインが所有する PCIe バスからのデバイスがあるすべての I/O ドメインに影響を与えます。[154 ページの「PCIe エンドポイントを構成した状態のルートドメインのリブート」](#)を参照してください。
- ルートドメインの SR-IOV 仮想機能または PCIe スロットを別のルートドメインに割り当てることはできません。この制限により、循環依存性が妨げられます。

primary 以外のルートドメインの例

次の例では、PCIe バスの I/O 仮想化を有効にし、primary 以外のルートドメインで直接 I/O デバイスを管理し、primary 以外のルートドメインで SR-IOV 仮想機能を管理する方法を説明します。

PCIe バスの I/O 仮想化の有効化

次の例に、`ldm add-io` および `ldm set-io` コマンドを使用して、I/O 仮想化を有効にする方法を示します。

次の SPARC T4-2 I/O 構成は `pci_1` が primary ドメインからすでに削除されていることを示しています。

```
primary# ldm list-io
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
pci_0                              BUS   pci_0   primary IOV
pci_1                              BUS   pci_1
niu_0                              NIU   niu_0   primary
niu_1                              NIU   niu_1   primary
/SYS/MB/PCIE0                      PCIE  pci_0   primary OCC
/SYS/MB/PCIE2                      PCIE  pci_0   primary OCC
/SYS/MB/PCIE4                      PCIE  pci_0   primary OCC
/SYS/MB/PCIE6                      PCIE  pci_0   primary EMP
/SYS/MB/PCIE8                      PCIE  pci_0   primary EMP
/SYS/MB/SASHBA                    PCIE  pci_0   primary OCC
/SYS/MB/NET0                      PCIE  pci_0   primary OCC
/SYS/MB/PCIE1                      PCIE  pci_1
/SYS/MB/PCIE3                      PCIE  pci_1
/SYS/MB/PCIE5                      PCIE  pci_1
/SYS/MB/PCIE7                      PCIE  pci_1
/SYS/MB/PCIE9                      PCIE  pci_1
/SYS/MB/NET2                      PCIE  pci_1
/SYS/MB/NET0/IOVNET.PF0            PF    pci_0   primary
/SYS/MB/NET0/IOVNET.PF1            PF    pci_0   primary
```

次のリストは、ゲストドメインがバインドされている状態であることを示しています。

```
primary# ldm list
NAME      STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary   active -n-cv-  UART   8     8G      0.6%  0.6%  8m
rootdom1  bound  ----- 5000   8     4G
ldg2      bound  ----- 5001   8     4G
ldg3      bound  ----- 5002   8     4G
```

次の `ldm add-io` コマンドは、`pci_1` バスを、そのバスに対して I/O 仮想化が有効になっている `rootdom1` ドメインに追加します。`ldm start` コマンドは `rootdom1` ドメインを起動します。

```
primary# ldm add-io iov=on pci_1 rootdom1
primary# ldm start rootdom1
LDom rootdom1 started
```

指定した PCIe バスがルートドメインにすでに割り当てられている場合は、`ldm set-io` コマンドを使用して、I/O 仮想化を有効にします。

```
primary# ldm start-reconf rootdom1
primary# ldm set-io iov=on pci_1
primary# ldm stop-domain -r rootdom1
```

I/O デバイスを構成する前に、ルートドメインでその OS が実行している必要があります。ゲストドメインがまだ自動ブートするように設定されていない場合、rootdom1 ゲストドメインのコンソールに接続し、rootdom1 ルートドメインの OS をブートします。

```
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Connecting to console "rootdom1" in group "rootdom1" ....
Press ~? for control options ..
ok> boot
...
primary#
```

次のコマンドは、pci_1 PCIe バスとその子が現在、rootdom1 ルートドメインによって所有されていることを示しています。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	rootdom1	IOV
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	rootdom1	OCC
/SYS/MB/PCIE3	PCIE	pci_1	rootdom1	OCC
/SYS/MB/PCIE5	PCIE	pci_1	rootdom1	OCC
/SYS/MB/PCIE7	PCIE	pci_1	rootdom1	OCC
/SYS/MB/PCIE9	PCIE	pci_1	rootdom1	EMP
/SYS/MB/NET2	PCIE	pci_1	rootdom1	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	rootdom1	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	rootdom1	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	rootdom1	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	rootdom1	

primary 以外のルートドメインでの直接 I/O デバイスの管理

次の例は、primary 以外のルートドメインで直接 I/O デバイスを管理する方法を示しています。

次のコマンドは、ルートドメインがアクティブである間に、ルートドメインからスロットを削除しようとしたため、エラーが生成されます。

```
primary# ldm remove-io /SYS/MB/PCIE7 ldg1
Dynamic I/O operations on PCIe slots are not supported.
Use start-reconf command to trigger delayed reconfiguration and make I/O
changes statically.
```

次のコマンドは、まずルートドメインで遅延再構成を開始して、スロットを削除する正しい方法を示しています。

```
primary# ldm start-reconf ldg1
Initiating a delayed reconfiguration operation on the ldg1 domain.
All configuration changes for other domains are disabled until the ldg1
domain reboots, at which time the new configuration for the ldg1 domain
will also take effect.
primary# ldm remove-io /SYS/MB/PCIE7 ldg1
-----
Notice: The ldg1 domain is in the process of a delayed reconfiguration.
Any changes made to the ldg1 domain will only take effect after it reboots.
-----
primary# ldm stop-domain -r ldg1
```

次の ldm list-io コマンドは、ルートドメインに /SYS/MB/PCIE7 スロットがなくなったことを確認します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	ldg1	IOV
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE3	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE5	PCIE	pci_1	ldg1	OCC

/SYS/MB/PCIE7	PCIE	pci_1	OCC
/SYS/MB/PCIE9	PCIE	pci_1	ldg1 EMP
/SYS/MB/NET2	PCIE	pci_1	ldg1 OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	ldg1
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	ldg1
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	ldg1
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	ldg1

次のコマンドは /SYS/MB/PCIE7 スロットを ldg2 ドメインに割り当てます。ldm start コマンドは ldg2 ドメインを起動します。

```
primary# ldm add-io /SYS/MB/PCIE7 ldg2
primary# ldm start ldg2
LDom ldg2 started
```

primary 以外のルートドメインでの SR-IOV 仮想機能の管理

これらのコマンドは、primary 以外のルートドメインに属している 2 つの物理機能のそれぞれから 2 つの仮想機能を作成します。

```
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF0
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF1
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
```

-n オプションを使用すれば、次の 2 つのコマンドで 2 つの仮想機能を作成することもできます。

```
primary# ldm create-vf -n 2 /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF1
primary# ldm create-vf -n 2 /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
```

指定した物理機能で仮想機能を動的に作成できなかった場合は、それらを静的に作成するために遅延再構成を開始します。

```
primary# ldm start-reconf ldg1
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
```

```

Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF0
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF1
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1

```

次に変更を有効にするために、ldg1 ルートドメインをリブートします。

```
primary# ldm stop-domain -r ldg1
```

次のコマンドは新しい仮想機能を表示します。

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
----	----	---	-----	-----
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	ldg1	IOV
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE3	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE5	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE7	PCIE	pci_1	ldg2	OCC
/SYS/MB/PCIE9	PCIE	pci_1	ldg1	EMP
/SYS/MB/NET2	PCIE	pci_1	ldg1	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF0.VF0	VF	pci_1		
/SYS/MB/PCIE5/IOVNET.PF0.VF1	VF	pci_1		
/SYS/MB/NET2/IOVNET.PF1.VF0	VF	pci_1		
/SYS/MB/NET2/IOVNET.PF1.VF1	VF	pci_1		

次のコマンドは、/SYS/MB/PCIE5/IOVNET.PF0.VF1 仮想関数を primary 以外のルートドメイン ldg1 に動的に追加します。

```
primary# ldm add-io /SYS/MB/PCIE5/IOVNET.PF0.VF1 ldg1
```

次のコマンドは、/SYS/MB/NET2/IOVNET.PF1.VF0 仮想機能を ldg2 ドメインに動的に追加します。

```
primary# ldm add-io /SYS/MB/NET2/IOVNET.PF1.VF0 ldg2
```

次のコマンドは、/SYS/MB/NET2/IOVNET.PF1.VF1 仮想機能を、バインドされた ldg3 ドメインに追加します。

```
primary# ldm add-io /SYS/MB/NET2/IOVNET.PF1.VF1 ldg3
primary# ldm start ldg3
LDom ldg3 started
```

ldg3 ドメインのコンソールに接続して、その OS をブートします。

次の出力は、すべての割り当てが想定したとおりに表示されていることを示しています。1 つの仮想機能が未割り当てであるため、ldg1、ldg2、または ldg3 ドメインにそれを動的に割り当てることができます。

```
# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	ldg1	IOV
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE3	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE5	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE7	PCIE	pci_1	ldg2	OCC
/SYS/MB/PCIE9	PCIE	pci_1	ldg1	EMP
/SYS/MB/NET2	PCIE	pci_1	ldg1	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF0.VF0	VF	pci_1		
/SYS/MB/PCIE5/IOVNET.PF0.VF1	VF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF1.VF0	VF	pci_1	ldg2	
/SYS/MB/NET2/IOVNET.PF1.VF1	VF	pci_1	ldg3	

◆◆◆ 第 10 章 10

仮想ディスクの使用

この章では、Oracle VM Server for SPARC ソフトウェアで仮想ディスクを使用する方法について説明します。

この章では、次の項目について説明します。

- [177 ページの「仮想ディスクの概要」](#)
- [179 ページの「仮想ディスクの識別子とデバイス名」](#)
- [180 ページの「仮想ディスクの管理」](#)
- [183 ページの「仮想ディスクの表示」](#)
- [184 ページの「仮想ディスクバックエンドオプション」](#)
- [186 ページの「仮想ディスクバックエンド」](#)
- [194 ページの「仮想ディスクマルチパスの構成」](#)
- [200 ページの「CD、DVD および ISO イメージ」](#)
- [203 ページの「仮想ディスクのタイムアウト」](#)
- [204 ページの「仮想ディスクおよび SCSI」](#)
- [205 ページの「仮想ディスクおよび format コマンド」](#)
- [205 ページの「仮想ディスクと ZFS の使用」](#)
- [210 ページの「Oracle VM Server for SPARC 環境でのボリュームマネージャーの使用」](#)
- [214 ページの「仮想ディスクに関する問題」](#)

仮想ディスクの概要

仮想ディスクには、2 つのコンポーネントがあります。ゲストドメインに表示される仮想ディスク自体と、データの格納先であり仮想 I/O の送信先である仮想ディスクバックエンドです。仮想ディスクバックエンドは、仮想ディスクサーバー (vds) ドライバによって、サービスドメインからエ

クスポートされます。vds ドライバは、論理ドメインチャネル (LDC) を使用して、ハイパーバイザを介してゲストドメインの仮想ディスククライアント (vdc) ドライバと通信します。最終的には、仮想ディスクはゲストドメイン内の /dev/[r]dsk/cXdYsZ デバイスとして表示されます。

注記 - ディスクパス名の一部として /dev/dsk または /dev/rdsk を使用することで、ディスクを参照できます。どちらの参照でも同じ結果になります。



注意 - ディスク全体を表すために d0 デバイスを使用しないでください。このデバイスは、ディスクに EFI ラベルがあり VTOC ラベルがない場合のみ、ディスク全体を表します。d0 デバイスを使用すると、仮想ディスクが単一スライスのディスクになり、ディスクの先頭に書き込みを行った場合にディスクラベルを破損してしまう可能性があります。

代わりに、s2 スライスを使用してディスク全体を仮想化します。s2 スライスはラベルとは無関係です。

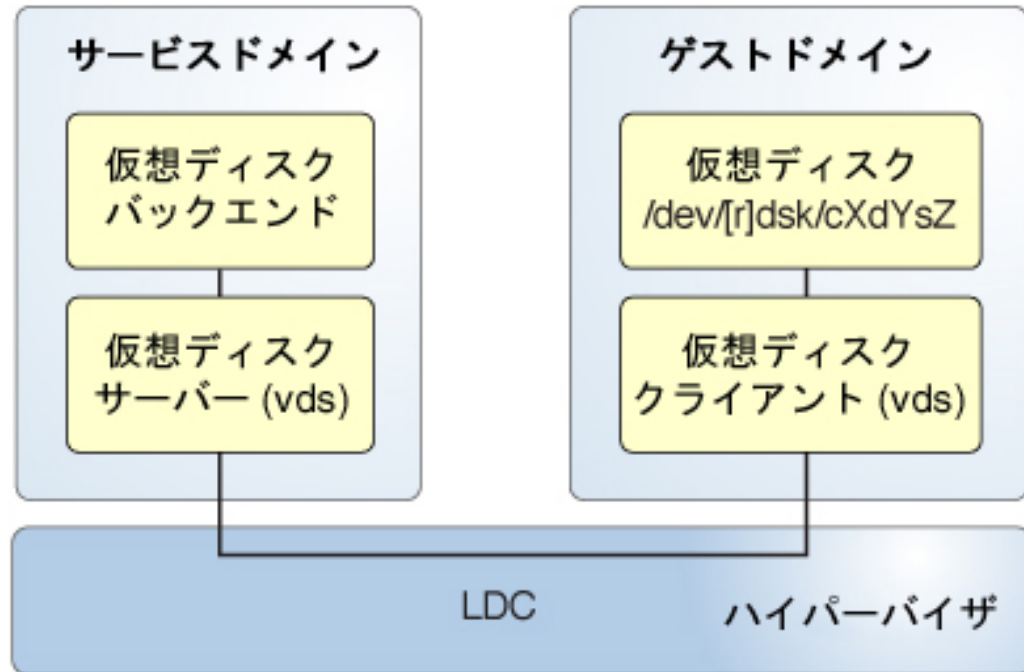
仮想ディスクバックエンドは、物理的でも論理的でもかまいません。物理デバイスには、次のものを含めることができます。

- 物理ディスクまたはディスク論理ユニット番号 (LUN)
- 物理ディスクスライス

論理デバイスは、次のいずれかにすることができます。

- ZFS や UFS などのローカルファイルシステム上または NFS によって使用可能なリモートファイルシステム上のファイル
- ZFS、VxVM、Solaris Volume Manager などのボリュームマネージャーからの論理ボリューム
- サービスドメインからアクセス可能な任意のディスク擬似デバイス

図 10-1 Oracle VM Server for SPARC を使用した仮想ディスク



仮想ディスクの識別子とデバイス名

`ldm add-vdisk` コマンドを使用してドメインに仮想ディスクを追加する際に、`id` プロパティを設定して、その仮想ディスクのデバイス番号を指定できます。

```
ldm add-vdisk [id=disk-id] disk-name volume-name@service-name domain-name
```

ドメインの各仮想ディスクには、ドメインがバインドされるときに割り当てられる一意のデバイス番号があります。`id` プロパティを設定して仮想ディスクを明示的なデバイス番号で追加した場合、指定したデバイス番号が使用されます。デバイス番号を指定しなかった場合、使用可能なもっとも小さいデバイス番号が自動的に割り当てられます。その場合、割り当てられるデバイス番号は、仮想ディスクがドメインに追加された方法によって異なります。仮想ディスクに最終的に割り当てられたデバイス番号は、ドメインがバインドされるときに `ldm list-bindings` コマンドの出力で確認できます。

仮想ディスクが構成されたドメインで Oracle Solaris OS を実行している場合、そのドメインでは、各仮想ディスクは `c0dn` ディスクデバイスとして表示されます。`n` は仮想ディスクのデバイス番号です。

次の例では、`ldg1` ドメインに、`rootdisk` と `pdisk` という 2 つの仮想ディスクがあります。`rootdisk` のデバイス番号は `0` (`disk@0`) で、ドメインではディスクデバイス `c0d0` として表示されます。`pdisk` のデバイス番号は `1` (`disk@1`) で、ドメインではディスクデバイス `c0d1` として表示されます。

```
primary# ldm list-bindings ldg1
...
DISK
      NAME                VOLUME                                TOUT DEVICE  SERVER      MPGROUP
      rootdisk            dsk_nevada@primary-vds0          disk@0  primary
      pdisk                c3t40d1@primary-vds0          disk@1  primary
...
```



注意 - デバイス番号が仮想ディスクに明示的に割り当てられていない場合、ドメインのバインドがいったん解除されたあとでふたたびバインドされると、デバイス番号が変更されることがあります。その場合、ドメインで実行している OS によって割り当てられたデバイス名も変更され、システムの既存の構成が損なわれることがあります。これは、たとえば、仮想ディスクがドメインの構成から削除されたときに起こる場合があります。

仮想ディスクの管理

このセクションでは、ゲストドメインへの仮想ディスクの追加、仮想ディスクオプションとタイムアウトオプションの変更、およびゲストドメインからの仮想ディスクの削除について説明します。仮想ディスクオプションの詳細については、[184 ページの「仮想ディスクバックエンドオプション」](#)を参照してください。仮想ディスクのタイムアウトの説明については、[203 ページの「仮想ディスクのタイムアウト」](#)を参照してください。

仮想ディスクバックエンドは、同じ仮想ディスクまたは別の仮想ディスクサーバーのいずれかを介して複数回エクスポートできます。仮想ディスクバックエンドのエクスポートされたインスタンスは、それぞれ同じゲストドメインまたは別のゲストドメインのいずれかに割り当てることができます。

仮想ディスクバックエンドを複数回エクスポートする場合は、排他 (`excl`) オプションを指定してエクスポートしないでください。`excl` オプションを指定すると、バックエンドのエクスポートは 1 回のみ許可されます。`ro` オプションを指定すると、バックエンドは読み取り専用デバイスとして問題なく複数回エクスポートできます。

仮想ディスクデバイスをドメインに割り当てると、仮想ディスクサービスを提供するドメインに対して暗黙的な依存関係が生まれます。ldm list-dependencies コマンドを使用することで、これらの依存関係を表示したり、仮想ディスクサービスに依存しているドメインを表示したりできます。424 ページの「ドメイン I/O の依存関係の一覧表示」を参照してください。

▼ 仮想ディスクを追加する方法

1. 仮想ディスクバックエンドをサービスドメインからエクスポートします。

```
ldm add-vdsdev [-fq] [options={ro,slice,excl}] [mpgroup=mpgroup] \
backend volume-name@service-name
```

2. このバックエンドをゲストドメインに割り当てます。

```
ldm add-vdisk [timeout=seconds] [id=disk-id] disk-name volume-name@service-name domain-name
```

id プロパティを設定して、新しい仮想ディスクデバイスのカスタム ID を指定できます。デフォルトでは ID 値は自動的に生成されるため、OS で既存のデバイス名に一致させる必要がある場合に、このプロパティを設定します。179 ページの「仮想ディスクの識別子とデバイス名」を参照してください。

注記 - バックエンドは、ゲストドメイン (domain-name) がバインドされたときに、実際にサービスドメインからエクスポートされ、ゲストドメインに割り当てられます。

▼ 仮想ディスクバックエンドを複数回エクスポートする方法



注意 - 仮想ディスクバックエンドが複数回エクスポートされる際は、ゲストドメインで動作中のアプリケーションおよびその仮想ディスクを使用中のアプリケーションが、同時の書き込みアクセスを調整および同期化して、データの一貫性を確保する役割を果たします。

次の例では、同じ仮想ディスクサービスを介して 2 つの異なるゲストドメインに同じ仮想ディスクを追加する方法について説明します。

1. サービスドメインから仮想ディスクバックエンドを 2 回エクスポートします。

```
ldm add-vdsdev [options={ro,slice}] backend volume1@service-name
# ldm add-vdsdev -f [options={ro,slice}] backend volume2@service-name
```

2 つめの `ldm add-vdsdev` コマンドでは、`-f` オプションを使用して、バックエンドの 2 回目のエクスポートを強制実行します。両方のコマンドに同じバックエンドパスを使用する場合や、仮想ディスクサーバーが同じサービスドメインに存在する場合に、このオプションを使用します。

2. エクスポートされたバックエンドを各ゲストドメインに割り当てます。

`ldom1` と `ldom2` には、異なる *disk-name* を指定できます。

```
ldm add-vdisk [timeout=seconds] disk-name volume1@service-name ldom1
# ldm add-vdisk [timeout=seconds] disk-name volume2@service-name ldom2
```

▼ 仮想ディスクオプションを変更する方法

仮想ディスクオプションの詳細については、[184 ページの「仮想ディスクバックエンドオプション」](#)を参照してください。

- サービスドメインからバックエンドがエクスポートされたあとに、仮想ディスクオプションを変更できます。

```
primary# ldm set-vdsdev options=[{ro,slice,excl}] volume-name@service-name
```

▼ タイムアウトオプションを変更する方法

仮想ディスクオプションの詳細については、[184 ページの「仮想ディスクバックエンドオプション」](#)を参照してください。

- 仮想ディスクがゲストドメインに割り当てられたあとに、仮想ディスクのタイムアウトを変更できます。

```
primary# ldm set-vdisk timeout=seconds disk-name domain-name
```

▼ 仮想ディスクを削除する方法

1. ゲストドメインから仮想ディスクを削除します。

```
primary# ldm rm-vdisk disk-name domain-name
```

2. サービスドメインから対応するバックエンドのエクスポートを停止します。

```
primary# ldm rm-vdsdev volume-name@service-name
```

仮想ディスクの表示

バックエンドが仮想ディスクとしてエクスポートされると、ゲストドメインにフルディスクまたは 1 つのスライスディスクとして表示可能になります。表示形式は、バックエンドの種類およびバックエンドのエクスポート時に使用したオプションによって異なります。



注意 - SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーでは、NVMe (Non-Volatile Memory Express) ストレージが導入されています。これは、ディスクドライブまたは Flash Accelerator F160 PCIe カードである場合があります。このディスクタイプをフルディスクとして使用して仮想ディスクバックエンドを構築することはできません。

このようなディスクを使用する場合は、`vdsdev` を作成するときにディスクのスライスまたは `slice` オプションを使用します。たとえば、ディスクをパーティション分割したあと、次のいずれかのコマンドを使用します。

```
ldm add-vdsdev dev/dsk/c14t1d0s6 volume3@primary-vds4
```

または:

```
ldm add-vdsdev options=slice /dev/dsk/c14t1d0s2 volume3@primary-vds4
```



注意 - 単一スライスのディスクはデバイス ID がありません。デバイス ID が必要な場合は、完全な物理ディスクのバックエンドを使用します。

フルディスク

バックエンドをフルディスクとしてドメインにエクスポートすると、そのドメインに、8 つのスライス (`s0 - s7`) を持つ通常のディスクとして表示されます。このタイプのディスクは、`format(1M)` コマンドを使用して表示できます。ディスクのパーティションテーブルは、`fmthard` コマンドまたは `format` コマンドのいずれかを使用して変更できます。

また、フルディスクは OS インストールソフトウェアからも表示でき、OS のインストール先のディスクとして選択できます。

どのバックエンドも、フルディスクとしてエクスポートできますが、1 つのスライスディスクとしてのみエクスポート可能な物理ディスクスライスは除きます。

1 つのスライスディスク

バックエンドを 1 つのスライスディスクとしてドメインにエクスポートすると、そのドメインに、8 つのスライス (s0 - s7) を持つ通常のディスクとして表示されます。ただし、使用できるのは 1 番目のスライス (s0) のみです。このタイプのディスクは、`format(1M)` コマンドで表示できますが、ディスクのパーティションテーブルは変更できません。

また、1 つのスライスディスクは OS インストールソフトウェアからも表示でき、OS のインストール先のディスクとして選択できます。この場合、UNIX ファイルシステム (UNIX File System、UFS) を使用して OS をインストールするときは、ルートパーティション (/) のみを定義し、このパーティションがすべてのディスク領域を使用する必要があります。

どのバックエンドも、1 つのスライスディスクとしてエクスポートできます。ただし、フルディスクとしてのみエクスポートできる物理ディスクは除きます。

注記 - Oracle Solaris 10 10/08 OS より前のリリースでは、1 つのスライスディスクは、1 つのパーティションを持つディスクとして表示されていました (s0)。このタイプのディスクは、`format` コマンドでは表示できませんでした。また、OS インストールソフトウェアからも表示できず、OS をインストール可能なディスクデバイスとして選択することができませんでした。

仮想ディスクバックエンドオプション

仮想ディスクのバックエンドをエクスポートする際には、さまざまなオプションを指定できます。これらのオプションは、`ldm add-vdsdev` コマンドの `options=` 引数にコンマ区切りのリストとして指定します。有効なオプションは、`ro`、`slice`、および `excl` です。

読み取り専用 (ro) オプション

読み取り専用 (ro) オプションは、バックエンドが読み取り専用デバイスとしてエクスポートされることを指定します。その場合、ゲストドメインに割り当てられるこの仮想ディスクに対しては読み取り操作のアクセスのみが可能で、仮想ディスクへの書き込み操作は失敗します。

排他 (excl) オプション

排他 (excl) オプションは、サービスマインのバックエンドを仮想ディスクとして別のドメインにエクスポートするときに、仮想ディスクサーバーによって排他的に開かれる必要があることを指定します。バックエンドが排他的に開かれると、サービスマインのほかのアプリケーションがこのバックエンドにアクセスすることはできません。この制限によって、サービスマインで動作するアプリケーションが、ゲストドメインでも使用されているバックエンドを誤って使用することはなくなります。

注記 - ドライバには excl オプションを受け入れないものもあるため、一部の仮想ディスクバックエンドを排他的に開くことが許可されません。excl オプションが物理ディスクおよびスライスで機能することはわかっていますが、このオプションはファイルでは機能しません。ディスクボリュームなどの擬似デバイスで機能できます。バックエンドのドライバで排他的オープンが受け入れられない場合、バックエンドの excl オプションは無視され、バックエンドは排他的に開かれません。

excl オプションによって、サービスマインで動作中のアプリケーションが、ゲストドメインにエクスポートされるバックエンドにアクセスできなくなるため、次の場合は excl オプションを設定しないでください。

- ゲストドメインの動作中に、format または luxadm などのコマンドを使用して物理ディスクを管理できるようにする場合は、excl オプションを指定してこれらのディスクをエクスポートしないでください。
- RAID またはミラー化ボリュームなどの Solaris Volume Manager ボリュームをエクスポートする場合は、excl オプションを設定しないでください。このようにしないと、RAID またはミラー化ボリュームのコンポーネントに障害が発生した場合に、Solaris Volume Manager で一部の復旧処理の開始が妨げられる可能性があります。詳細については、[211 ページの「Solaris ボリュームマネージャーによる仮想ディスクの使用」](#)を参照してください。
- Veritas Volume Manager (VxVM) がサービスマインにインストールされていて、Veritas Dynamic Multipathing (VxDMP) が物理ディスクに対して有効な場合は、excl オプション (デフォルトではない) を指定せずに物理ディスクをエクスポートする必要があります。このようにしないと、仮想ディスクサーバー (vds) が物理ディスクデバイスを開くことができないため、エクスポートは失敗します。詳細は、[212 ページの「VxVM のインストール時の仮想ディスクの使用」](#)を参照してください。
- 同じ仮想ディスクバックエンドを同じ仮想ディスクサービスから複数回エクスポートする場合の詳細については、[181 ページの「仮想ディスクバックエンドを複数回エクスポートする方法」](#)を参照してください。

デフォルトでは、バックエンドは排他的ではない状態で開かれます。このため、バックエンドが別のドメインにエクスポートされている間でも、サービストメインで動作中のアプリケーションはこのバックエンドを使用できます。

スライス (slice) オプション

通常、バックエンドは、その種類に応じてフルディスクまたは 1 つのスライスディスクのいずれかとしてエクスポートされます。slice オプションを指定すると、バックエンドは強制的に 1 つのスライスディスクとしてエクスポートされます。

このオプションは、バックエンドの raw コンテンツをエクスポートする場合に便利です。たとえば、データを格納済みの ZFS または Solaris Volume Manager ボリュームがある場合に、ゲストドメインでこのデータにアクセスするには、slice オプションを使用して ZFS または Solaris Volume Manager ボリュームをエクスポートする必要があります。

このオプションの詳細については、[186 ページの「仮想ディスクバックエンド」](#)を参照してください。

仮想ディスクバックエンド

仮想ディスクバックエンドは、仮想ディスクのデータの格納場所です。バックエンドには、ディスク、ディスクスライス、ファイル、またはボリューム (ZFS、Solaris Volume Manager、VxVM など) を使用できます。バックエンドは、バックエンドをサービストメインからエクスポートする際に slice オプションを設定するかどうかに応じて、フルディスクまたは 1 つのスライスディスクのいずれかとしてゲストドメインに表示されます。デフォルトでは、仮想ディスクバックエンドは読み取りおよび書き込み可能なフルディスクとして排他的でない状態でエクスポートされます。

物理ディスクまたはディスクの LUN

物理ディスクまたはディスク LUN は、常にフルディスクとしてエクスポートされます。この場合、仮想ディスクドライバ (vds および vdc) は仮想ディスクからの入出力を転送し、物理ディスクまたはディスク LUN へのパススルーとして動作します。

`slice` オプションを設定せずにそのディスクのスライス 2 (`s2`) に対応するデバイスをエクスポートすると、物理ディスクまたはディスク LUN はサービスドメインからエクスポートされます。`slice` オプションを指定してディスクのスライス 2 をエクスポートすると、ディスク全体ではなくこのスライスのみがエクスポートされます。

▼ 物理ディスクを仮想ディスクとしてエクスポートする方法



注意 - 仮想ディスクを構成する際には、各仮想ディスクが物理ディスク、ディスクスライス、ファイル、ボリュームなどの個別の物理 (バックエンド) リソースを参照していることを確認してください。

FibreChannel や SAS などの一部のディスクは、2 つの異なるパスから同じディスクを参照できることを意味する「デュアルポート」の特徴を備えています。異なるドメインに割り当てられたパスが同じ物理ディスクを参照していないことを確認してください。

1. 物理ディスクを仮想ディスクとしてエクスポートします。

たとえば、物理ディスク `c1t48d0` を仮想ディスクとしてエクスポートするには、そのディスクのスライス 2 (`c1t48d0s2`) をエクスポートする必要があります。

```
primary# ldm add-vdsdev /dev/dsk/c1t48d0s2 c1t48d0@primary-vds0
```

2. このディスクをゲストドメインに割り当てます。

たとえば、ディスク `pdisk` をゲストドメイン `ldg1` に割り当てます。

```
primary# ldm add-vdisk pdisk c1t48d0@primary-vds0 ldg1
```

3. ゲストドメインが起動されて Oracle Solaris OS が実行されたら、そのディスクがアクセス可能であり、フルディスクであることを確認します。

フルディスクとは、8 つのスライスを持つ通常のディスクのことです。

確認するディスクが `c0d1` の場合、次のようになります。

```
ldg1# ls -l /dev/dsk/c0d1s*
/dev/dsk/c0d1s0
/dev/dsk/c0d1s1
/dev/dsk/c0d1s2
/dev/dsk/c0d1s3
/dev/dsk/c0d1s4
/dev/dsk/c0d1s5
/dev/dsk/c0d1s6
```

```
/dev/dsk/c0d1s7
```

物理ディスクスライス

物理ディスクスライスは、常に 1 つのスライスディスクとしてエクスポートされます。この場合、仮想ディスクドライバ (vds および vdc) は仮想ディスクから入出力を転送し、物理ディスクスライスへのパススルーとして動作します。

物理ディスクスライスは、対応するスライスデバイスをエクスポートすることで、サービドメインからエクスポートされます。デバイスがスライス 2 と異なる場合は、slice オプションの指定の有無にかかわらず、自動的に 1 つのスライスディスクとしてエクスポートされます。デバイスがディスクのスライス 2 である場合は、slice オプションを設定して、スライス 2 のみを 1 つのスライスディスクとしてエクスポートする必要があります。そうしないと、ディスク全体がフルディスクとしてエクスポートされます。



注意 - SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーでは、NVMe (Non-Volatile Memory Express) ストレージが導入されています。これは、ディスクドライブまたは Flash Accelerator F160 PCIe カードである場合があります。このディスクタイプをフルディスクとして使用して仮想ディスクバックエンドを構築することはできません。

このようなディスクを使用する場合は、vdsdev を作成するときにディスクのスライスまたは slice オプションを使用します。

▼ 物理ディスクスライスを仮想ディスクとしてエクスポートする方法

1. 物理ディスクのスライスを仮想ディスクとしてエクスポートします。

たとえば、物理ディスク c1t57d 0 のスライス 0 を仮想ディスクとしてエクスポートするには、そのスライス (c1t57d0s0) に対応するデバイスを次のようにエクスポートする必要があります。

```
primary# ldm add-vdsdev /dev/dsk/c1t57d0s0 c1t57d0s0@primary-vds0
```

スライスは常に 1 つのスライスディスクとしてエクスポートされるため、slice オプションを指定する必要はありません。

2. このディスクをゲストドメインに割り当てます。

たとえば、ディスク pslice をゲストドメイン ldg1 に割り当てます。

```
primary# ldm add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

3. ゲストメインが起動されて Oracle Solaris OS が実行されたら、ディスク (c0d13 など) を表示して、そのディスクがアクセス可能であることを確認できます。

```
ldg1# ls -l /dev/dsk/c0d13s*
/dev/dsk/c0d13s0
/dev/dsk/c0d13s1
/dev/dsk/c0d13s2
/dev/dsk/c0d13s3
/dev/dsk/c0d13s4
/dev/dsk/c0d13s5
/dev/dsk/c0d13s6
/dev/dsk/c0d13s7
```

デバイスは 8 つありますが、そのディスクは 1 つのスライスディスクであるため、使用できるのは 1 番目のスライス (s0) のみです。

▼ スライス 2 をエクスポートする方法

- スライス 2 (ディスク c1t57d0s2 など) をエクスポートするには、slice オプションを指定する必要があります。そうしないと、ディスク全体がエクスポートされます。

```
primary# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

ファイルおよびボリュームのエクスポート

ファイルまたはボリューム (たとえば ZFS または Solaris Volume Manager からの) は、slice オプションの指定の有無に応じて、フルディスクまたは 1 つのスライスディスクのいずれかとしてエクスポートされます。

フルディスクとしてエクスポートされるファイルまたはボリューム

slice オプションを設定しない場合、ファイルまたはボリュームはフルディスクとしてエクスポートされます。この場合、仮想ディスクドライバ (vds および vdc) は仮想ディスクから入出力を転送し、仮想ディスクのパーティション分割を管理します。最終的には、このファイルまたはボリュームは、仮想ディスクのすべてのスライスのデータ、およびパーティション分割とディスク構造の管理に使用されるメタデータを含むディスクイメージになります。

空のファイルまたはボリュームをフルディスクとしてエクスポートすると、未フォーマットのディスク、つまり、パーティションのないディスクとしてゲストドメインに表示されます。このため、ゲストドメインで `format` コマンドを実行して、使用可能なパーティションを定義し、有効なディスクラベルを書き込む必要があります。ディスクが未フォーマットの間、この仮想ディスクへの入出力はすべて失敗します。

注記 - パーティションを作成するには、ゲストドメインで `format` コマンドを実行する必要があります。

▼ ファイルをフルディスクとしてエクスポートする方法

1. サービスドメインから、ファイル (`fdisk0` など) を作成して仮想ディスクとして使用します。

```
service# mkfile 100m /ldoms/domain/test/fdisk0
```

ファイルのサイズによって、仮想ディスクのサイズが定義されます。この例では、100M バイトの空のファイルを作成して、100M バイトの仮想ディスクを取得しています。

2. 制御ドメインから、ファイルを仮想ディスクとしてエクスポートします。

```
primary# ldm add-vdsdev /ldoms/domain/test/fdisk0 fdisk0@primary-vds0
```

この例では、`slice` オプションを設定していないため、ファイルはフルディスクとしてエクスポートされます。

3. 制御ドメインから、ディスクをゲストドメインに割り当てます。

たとえば、ディスク `fdisk` をゲストドメイン `ldg1` に割り当てます。

```
primary# ldm add-vdisk fdisk fdisk0@primary-vds0 ldg1
```

4. ゲストドメインが起動されて Oracle Solaris OS が実行されたら、そのディスクがアクセス可能であり、フルディスクであることを確認します。

フルディスクとは、8 つのスライスを持つ通常のディスクのことです。

次の例は、ディスク `c0d5` を表示して、そのディスクがアクセス可能であり、フルディスクであることを確認する方法を示しています。

```
ldg1# ls -l /dev/dsk/c0d5*
/dev/dsk/c0d5s0
/dev/dsk/c0d5s1
/dev/dsk/c0d5s2
/dev/dsk/c0d5s3
```

```
/dev/dsk/c0d5s4
/dev/dsk/c0d5s5
/dev/dsk/c0d5s6
/dev/dsk/c0d5s7
```

▼ ZFS ボリュームをフルディスクとしてエクスポートする方法

1. フルディスクとして使用する ZFS ボリュームを作成します。

次の例は、フルディスクとして使用する ZFS ボリューム `zdisk0` を作成する方法を示しています。

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

ボリュームのサイズによって、仮想ディスクのサイズが定義されます。この例では、結果的に 100M バイトの仮想ディスクになる、100M バイトのボリュームを作成します。

2. 制御ドメインから、その ZFS ボリュームに対応するデバイスをエクスポートします。

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

この例では、`slice` オプションを設定していないため、ファイルはフルディスクとしてエクスポートされます。

3. 制御ドメインから、ボリュームをゲストドメインに割り当てます。

次の例は、ボリューム `zdisk0` をゲストドメイン `ldg1` に割り当てる方法を示しています。

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4. ゲストドメインが起動されて Oracle Solaris OS が実行されたら、そのディスクがアクセス可能であり、フルディスクであることを確認します。

フルディスクとは、8 つのスライスを持つ通常のディスクのことです。

次の例は、ディスク `c0d9` を表示して、そのディスクがアクセス可能であり、フルディスクであることを確認する方法を示しています。

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7
```

1 つのスライスディスクとしてエクスポートされるファイルまたはボリューム

`slice` オプションを設定すると、ファイルまたはボリュームは 1 つのスライスディスクとしてエクスポートされます。この場合、仮想ディスクには 1 つのパーティション (`s0`) のみが含まれ、このパーティションが直接ファイルまたはボリュームバックエンドにマップされます。ファイルまたはボリュームには仮想ディスクに書き込まれるデータのみが含まれ、パーティション情報やディスク構造などの追加データは含まれません。

ファイルまたはボリュームが 1 つのスライスディスクとしてエクスポートされると、システムは擬似的なディスクのパーティション分割のシミュレーションを行います。これにより、そのファイルまたはボリュームはディスクスライスとして表示されます。ディスクのパーティション分割のシミュレーションが行われるため、そのディスクに対してパーティションは作成しないでください。

▼ ZFS ボリュームを 1 つのスライスディスクとしてエクスポートする方法

1. ZFS ボリュームを作成して、1 つのスライスディスクとして使用します。

次の例は、ZFS ボリューム `zdisk0` を作成して、1 つのスライスディスクとして使用する方法を示しています。

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

ボリュームのサイズによって、仮想ディスクのサイズが定義されます。この例では、100M バイトのボリュームを作成して、100M バイトの仮想ディスクを取得します。

2. 制御ドメインから、その ZFS ボリュームに対応するデバイスをエクスポートします。このボリュームが 1 つのスライスディスクとしてエクスポートされるように `slice` オプションを設定します。

```
primary# ldm add-vdsdev options=slice /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

3. 制御ドメインから、ボリュームをゲストドメインに割り当てます。

次の例は、ボリューム `zdisk0` をゲストドメイン `ldg1` に割り当てる方法を示しています。

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4. ゲストドメインが起動されて Oracle Solaris OS が実行されたら、ディスク (`c0d9` など) を表示して、そのディスクがアクセス可能で、1 つのスライスディスク (`s0`) であることを確認できます。

```
ldg1# ls -l /dev/dsk/c0d9s*
```



```

/dev/dsk/c0d9s0
/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7

```

ボリュームのエクスポートおよび下位互換性

ボリュームを仮想ディスクとしてエクスポートする構成がある場合、ボリュームは 1 つのスライスディスクではなくフルディスクとしてエクスポートされるようになります。アップグレード前の動作を保持して、ボリュームを 1 つのスライスディスクとしてエクスポートするには、次のいずれかを実行する必要があります。

- Oracle VM Server for SPARC 3.3 ソフトウェアで `ldm set-vdsdev` コマンドを使用して、1 つのスライスディスクとしてエクスポートするすべてのボリュームに `slice` オプションを設定します。[ldm\(1M\)](#) マニュアルページを参照してください。
- 次の行を、サービスドメインの `/etc/system` ファイルに追加します。

```
set vds:vd_volume_force_slice = 1
```

`/etc/system` プロパティ値の正しい作成または更新については、[400 ページの「`/etc/system` ファイルのプロパティ値の更新」](#)を参照してください。

注記 - この調整可能なオプションを設定すると、すべてのボリュームが強制的に 1 つのスライスディスクとしてエクスポートされ、ボリュームをフルディスクとしてエクスポートできなくなります。

各種のバックエンドのエクスポート方法のサマリー

バックエンド	スライスオプションなし	スライスオプションを設定
ディスク (ディスクスライス 2)	フルディスク [†]	1 つのスライスディスク [‡]
ディスクスライス (スライス 2 以外)	1 つのスライスディスク [*]	1 つのスライスディスク
ファイル	フルディスク	1 つのスライスディスク

バックエンド	スライスオプションなし	スライスオプションを設定
ボリューム (ZFS, Solaris Volume Manager, VxVM を含む)	フルディスク	1 つのスライスディスク

† ディスク全体をエクスポートします。

‡ スライス 2 のみをエクスポートします。

* スライスは常に 1 つのスライスディスクとしてエクスポートされます。

ファイルおよびディスクスライスを仮想ディスクとしてエクスポートする場合のガイドライン

このセクションでは、ファイルおよびディスクスライスを仮想ディスクとしてエクスポートする場合のガイドラインを示します。

ループバックファイル (lofi) ドライバの使用

ファイルを仮想ディスクとしてエクスポートするループバックファイル (lofi) ドライバを使用すると、余分なドライバレイヤーが追加されるため、仮想ディスクのパフォーマンスに影響します。代わりに、フルディスクまたは 1 つのスライスディスクとしてファイルを直接エクスポートすることができます。[189 ページの「ファイルおよびボリュームのエクスポート」](#)を参照してください。

ディスクスライスの直接的または間接的なエクスポート

仮想ディスクとしてスライスを直接的に、または Solaris Volume Manager ボリュームを介するなどして間接的にエクスポートするには、prtvtoc コマンドを使用して、スライスが物理ディスクの最初のブロック (ブロック 0) で開始されていないことを確認します。

物理ディスクの最初のブロックから始まるディスクスライスを直接的または間接的にエクスポートする場合は、物理ディスクのパーティションテーブルを上書きして、そのディスクのすべてのパーティションにアクセスできないようにすることもできます。

仮想ディスクマルチパスの構成

仮想ディスクマルチパスを使用すると、ゲストドメインに仮想ディスクを構成して、複数のパスからバックエンドストレージにアクセスできます。それらのパスは、ディスク LUN などの同一のバツ

クエンドストレージにアクセスできる各種サービスドメインを通過します。この機能では、サービスドメインのいずれかがダウンしても、ゲストドメイン内の仮想ディスクをアクセス可能なままにしておくことができます。たとえば、仮想ディスクマルチパス構成を設定して、ネットワークファイルシステム (Network File System、NFS) サーバーのファイルにアクセスすることがあります。また、この構成を使用して、複数のサービスドメインに接続される共有ストレージから LUN にアクセスできます。このため、ゲストドメインが仮想ディスクにアクセスする場合、仮想ディスクドライバは、サービスドメインのいずれかを通過してバックエンドストレージにアクセスします。仮想ディスクドライバがサービスドメインに接続できない場合、仮想ディスクは、別のサービスドメインを通じてバックエンドストレージにアクセスします。

注記 - mpgroups と SCSI 予約を一緒に使用することはできません。

仮想ディスクマルチパス機能では、サービスドメインがバックエンドストレージにアクセスできない場合を検出できます。そのような場合、仮想ディスクドライバは、別のパスでバックエンドストレージへのアクセスを試みます。

仮想ディスクマルチパスを有効にするには、各サービスドメインから仮想ディスクバックエンドをエクスポートし、同じマルチパスグループ (mpgroup) に追加する必要があります。仮想ディスクバックエンドがエクスポートされると、mpgroup は名前で識別され、構成されます。

次の図は、[197 ページの「仮想ディスクマルチパスを構成する方法」](#)の手順の例として使用されている仮想ディスクマルチパス構成を示しています。この例では、mpgroup1 というマルチパスグループを使用して仮想ディスクを作成しています。そのバックエンドには、primary と alternate という 2 つのサービスドメインからアクセスできます。

▼ 仮想ディスクマルチパスを構成する方法

図10-2「仮想ディスクマルチパスの構成」を参照してください。

1. 仮想ディスクバックエンドを **primary** サービスドメインからエクスポートします。

```
primary# ldm add-vdsdev mpgroup=mpgroup1 backend-path1 volume@primary-vds0
```

backend-path1 は、primary ドメインからの仮想ディスクバックエンドへのパスです。

2. 同じ仮想ディスクバックエンドを **alternate** サービスドメインからエクスポートします。

```
primary# ldm add-vdsdev mpgroup=mpgroup1 backend-path2 volume@alternate-vds0
```

backend-path2 は、alternate サービスドメインからの仮想ディスクバックエンドへのパスです。

注記 - *backend-path1* および *backend-path2* は、同じ仮想ディスクバックエンドのパスですが、それらのエクスポート元は異なる 2 つのドメイン (primary と alternate) です。これらのパスは、primary サービスドメインおよび alternate サービスドメインの構成に応じて、同じ場合もあれば、異なる場合もあります。*volume* 名はユーザーが選択します。これは、両方のコマンドで同じ場合もあれば、異なる場合もあります。

3. 仮想ディスクをゲストドメインにエクスポートします。

```
primary# ldm add-vdisk disk-name volume@primary-vds0 domain-name
```

注記 - 仮想ディスクバックエンドを複数のサービスドメインを介して複数回エクスポートしていますが、ゲストドメインに割り当てて、いずれかのサービスドメインを介して仮想ディスクバックエンドに関連付ける仮想ディスクは 1 つのみです。

例 10-1 Mpgroup を使用してプライマリドメインと代替ドメインの両方の仮想ディスクサービスに LUN を追加

次は、LUN を作成し、同じ mpgroup を使用してプライマリドメインと代替ドメインの両方の仮想ディスクサービスにそれを追加する方法を示しています。

LUN にアクセスするときにどちらのドメインを最初に使用するかを決定するには、ドメインにディスクを追加するときに、関連するパスを指定します。

- 仮想ディスクデバイスを作成します。

```
primary# ldm add-vdsdev mpgroup=ha lun1@primary-vds0
```

```
primary# ldm add-vdsdev mpgroup=ha lun1@alternate-vds0
```

- primary-vds0 からの LUN を最初に使用するには、次のコマンドを実行します。

```
primary# ldm add-vdisk disk1 lun1@primary-vds0 gd0
```

- alternate-vds0 からの LUN を最初に使用するには、次のコマンドを実行します。

```
primary# ldm add-vdisk disk1 lun1@alternate-vds0 gd0
```

仮想ディスクマルチパスの結果

仮想ディスクをマルチパスで構成し、ゲストドメインを起動すると、仮想ディスクは関連付けられているいずれかのサービスドメインを介してバックエンドにアクセスします。このサービスドメインが利用できなくなると、仮想ディスクは、同じマルチパスグループに属する別のサービスドメインを介してバックエンドへのアクセスを試みます。



注意 - マルチパスグループ (mpgroup) を定義する場合、同じ mpgroup に属する仮想ディスクバックエンドは、事実上同じ仮想ディスクバックエンドにする必要があります。異なるバックエンドを同じ mpgroup に追加すると、予期しない動作が生じ、それらのバックエンドに格納されているデータが消失または破損する可能性があります。

動的なパス選択

少なくとも Oracle Solaris 11.2 SRU 1 OS を実行しているゲストドメイン上の仮想ディスクに使用されるパスを動的に選択できます。

動的なパス選択は、ldm set-vdisk コマンドを使用して volume プロパティに *volume-name@service-name* という形式の値を設定することによって mpgroup ディスク内の最初のパスが変更された場合に発生します。動的なパス選択をサポートするアクティブドメインは、選択されたパスのみに切り替えることができます。更新されたドライバが実行されていない場合は、Oracle Solaris OS がディスクインスタンスをリロードしたとき、または次回のドメインリブート時にこのパスが選択されます。

動的なパス選択機能では、ディスクの使用中に次の手順を動的に実行できます。

- ディスクの接続時にゲストドメインによって最初に試行されるディスクパスを指定します。
- 現在アクティブなパスを、すでに接続済みのマルチパスディスク用に示されたパスに変更します。

mpgroup ディスクとともに ldm add-vdisk コマンドを使用すると、*volume-name@service-name* によって示されたパスが、ディスクにアクセスするために選択されたパスとして指定されます。

選択されたディスクパスは、関連する mpgroup が作成されたときのランクに関係なく、ゲストドメインに提供されるパスのセットの最初にリストされます。

ldm set-vdisk コマンドは、バインドされたドメイン、非アクティブなドメイン、およびアクティブなドメインで使用できます。アクティブなドメインで使用する場合、このコマンドでは、mpgroup ディスクの選択されたパスのみを選択できます。

ldm list-bindings コマンドは次の情報を表示します。

- 各 mpgroup パスの STATE 列には、次のいずれかの値が表示されます。
 - active – mpgroup の現在のアクティブなパスです。
 - standby – パスは現在使用されていません。
 - unknown – ドメインが動的なパス選択をサポートしていないか、デバイスが接続されていないか、エラーによってパスの状態を取得できません。
- ディスクパスは、アクティブなパスの選択に使用される順序でリストされます。
- ディスクに関連付けられたボリュームは mpgroup の選択されたパスであるため、最初に表示されます。

次の例は、選択されたパスは vol-ldg2@opath-ldg2 であり、現在使用されているアクティブなパスは ldg1 ドメインを通っていることを示しています。この状況は、選択されたパスが使用できず、2 番目に可能なパスが代わりに使用された場合に発生する可能性があります。選択されたパスがオンラインに復帰しても、選択されたパス以外のパスが引き続き使用されます。最初のパスを再度アクティブにするには、ldm set-vdisk コマンドを再発行して、volume プロパティに必要なパスの名前を設定します。

DISK

NAME	VOLUME	TOUT	ID	DEVICE	SERVER	MPGROUP
disk	disk-ldg4@primary-vds0	0	disk@0	primary		
tdiskgroup	vol-ldg2@opath-ldg2	1	disk@1	ldg2		testdiskgroup
PORT	MPGROUP	VOLUME	MPGROUP	SERVER	STATE	
2	vol-ldg2@opath-ldg2	ldg2			standby	
0	vol-ldg1@opath-vds	ldg1			active	
1	vol-prim@primary-vds0	primary			standby	

Oracle Solaris 11.2 SRU 1 OS 以上が実行されていないバインドされたドメインの mpgroup ディスクに対して ldm set-vdisk コマンドを使用した場合は、その操作によってパスの優先順位が変更されるため、次のディスク接続またはリブート中に、あるいは OBP からのアクセスが必要になった場合に新しいパスを最初に使用できます。

CD、DVD および ISO イメージ

コンパクトディスク (Compact Disc、CD) またはデジタル多用途ディスク (Digital Versatile Disc、DVD) のエクスポートは、通常のディスクと同じ方法で実行できます。CD または DVD をゲストドメインにエクスポートするには、CD または DVD デバイスのスライス 2 をフルディスクとして、つまり `slice` オプションを指定しないでエクスポートします。

注記 - CD または DVD ドライブ自体をエクスポートすることはできません。CD または DVD ドライブの中にある CD または DVD のみをエクスポートできます。このため、CD または DVD はエクスポート前にドライブ内に存在している必要があります。また、CD または DVD をエクスポートできるようにするには、その CD または DVD がサービスドメインで使用中心になっていない必要があります。特に、ボリューム管理ファイルシステムの `volfs` サービスが CD または DVD を使用してはいけません。`volfs` によるデバイスの使用を解除する方法については、[201 ページの「CD または DVD をサービスドメインからゲストドメインにエクスポートする方法」](#)を参照してください。

ファイルまたはボリュームに CD または DVD の ISO (国際標準化機構) イメージが格納されている場合に、そのファイルまたはボリュームをフルディスクとしてエクスポートすると、ゲストドメインで CD または DVD として表示されます。

CD、DVD、または ISO イメージをエクスポートすると、自動的にゲストドメインで読み取り専用デバイスとして表示されます。ただし、ゲストドメインから CD の制御操作を実行することはできません。つまり、ゲストドメインから CD の起動、停止、または取り出しは実行できません。エクスポートされた CD、DVD、または ISO イメージをブート可能な場合は、対応する仮想ディスクでゲストドメインをブートできます。

たとえば、Oracle Solaris OS インストール DVD をエクスポートした場合は、その DVD に対応する仮想ディスク上のゲストドメインをブートし、その DVD からゲストドメインをインストールすることができます。これを行うには、ゲストドメインで `ok` プロンプトが表示されたときに次のコマンドを使用します。

```
ok boot /virtual-devices@100/channel-devices@200/disk@n:f
```

`n` は、エクスポートされた DVD を表す仮想ディスクのインデックスです。

注記 - Oracle Solaris OS インストール DVD をエクスポートし、その DVD に対応する仮想ディスク上でゲストドメインをブートしてゲストドメインをインストールする場合、インストール中に DVD を変更することはできません。このため、異なる CD または DVD をリクエストするインストール手順は省略する必要がある場合があります。または、リクエストされたメディアにアクセスするための代替パスを指定する必要があります。

▼ CD または DVD をサービドメインからゲストドメインに エクスポートする方法

1. サーブドメインから、ボリューム管理デーモン `vold` が動作中でオンラインかどうかを確認します。

```
service# svcs volfs
STATE          STIME      FMRI
online         12:28:12  svc:/system/filesystem/volfs:default
```

- 手順 1 の例に示すように、ボリューム管理デーモンが動作中でオンラインの場合は、次の手順を実行します。

- a. `/etc/vold.conf` ファイルで、次の文字列で始まる行をコメントアウトします。

```
use cdrom drive....
```

vold.conf(4) マニュアルページを参照してください。

- b. CD または DVD ドライブに CD または DVD を挿入します。

- c. サービスドメインから、ボリューム管理ファイルシステムサービスを再起動します。

```
service# svcadm refresh volfs
service# svcadm restart volfs
```

3. サービスドメインから、CD-ROM デバイスのディスクパスを検出します。

```

service# cdrw -l
Looking for CD devices...

```

Node	Connected Device	Device type
/dev/rdskc/l1t0d0s2	MATSHITA CD-RW CW-8124	DZ13 CD Reader/Writer

4. CD または DVD ディスクデバイスをフルディスクとしてエクスポートします。

```
primary# ldm add-vdsdev /dev/dsk/c1t0d0s2 cdrom@primary-vds0
```

5. エクスポートされた CD または DVD をゲストドメインに割り当てます。

次のコマンドは、エクスポートされた CD または DVD をドメイン `ldg1` に割り当てる方法を示しています。

```
primary# ldm add-vdisk cdrom cdrom@primary-vds0 ldg1
```

CD または DVD の複数回のエクスポート

CD または DVD を複数回エクスポートして、さまざまなゲストドメインに割り当てることができます。詳細については、[181 ページの「仮想ディスクバックエンドを複数回エクスポートする方法」](#)を参照してください。

▼ 制御ドメインから ISO イメージをエクスポートしてゲストドメインをインストールする方法

始める前に この手順では、primary ドメインとゲストドメインの両方が構成されていることを前提としています。

たとえば、次のように `ldm list` を実行すると、primary ドメインと `ldom1` ドメインの両方が構成されていることが表示されます。

```
primary# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv  SP    8     8G      0.3%  15m
ldom1         active -t---  5000  4     1G      25%   8m
```

1. 仮想ディスクサーバーデバイスを追加して、ISO イメージをエクスポートします。

この例では、ISO イメージは `/export/images/sol-10-u8-ga-sparc-dvd.iso` です。

```
primary# ldm add-vdsdev /export/images/sol-10-u8-ga-sparc-dvd.iso dvd-iso@primary-vds0
```

2. ゲストドメインを停止します。

この例では、論理ドメインは `ldom1` です。

```
primary# ldm stop-domain ldom1
LDom ldom1 stopped
```

3. ISO イメージの仮想ディスクを論理ドメインに追加します。

この例では、論理ドメインは `ldom1` です。

```
primary# ldm add-vdisk s10-dvd dvd-iso@primary-vds0 ldom1
```

4. ゲストドメインを再起動します。

この例では、論理ドメインは `ldom1` です。

```
primary# ldm start-domain ldom1
LDom ldom1 started
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv	SP	8	8G	0.4%	25m
ldom1	active	-t---	5000	4	1G	0.0%	0s

この例では、`ldm list` コマンドにより、`ldom1` ドメインが起動されたばかりであることが表示されています。

5. ゲストドメインに接続します。

```
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldom1" in group "ldom1" ....
Press ~? for control options ..
```

6. ISO イメージが仮想ディスクとして追加されていることを確認します。

```
{0} ok show-disks
a) /virtual-devices@100/channel-devices@200/disk@1
b) /virtual-devices@100/channel-devices@200/disk@0
q) NO SELECTION
Enter Selection, q to quit: q
```

この例では、新しく追加されたデバイスは `/virtual-devices@100/channel-devices@200/disk@1` です。

7. ゲストドメインをブートして、ISO イメージからインストールします。

この例では、`/virtual-devices@100/channel-devices@200/disk@1` ディスクの `f` スライスからブートします。

```
{0} ok boot /virtual-devices@100/channel-devices@200/disk@1:f
```

仮想ディスクのタイムアウト

デフォルトでは、仮想ディスクバックエンドへのアクセスを提供するサービスドメインが停止すると、ゲストドメインから対応する仮想ディスクへのすべての入出力がブロックされます。サービスドメインが動作していて、仮想ディスクバックエンドへの入出力要求が処理されている場合、入出力は自動的に再開されます。

ただし、サービスドメインの停止状態が長すぎる場合に、ファイルシステムまたはアプリケーションにとって、入出力処理がブロックされるよりも、失敗してエラーが報告される方が望ましい場

合もあります。仮想ディスクごとに接続タイムアウト時間を設定できるようになったため、それをゲストドメインの仮想ディスククライアントとサービスドメインの仮想ディスクサーバー間の接続確立に使用できます。タイムアウト時間に達した場合、サービスドメインが停止し、仮想ディスククライアントと仮想ディスクサーバー間の接続が再確立されていない間中、保留中の入出力および新規の入出力は失敗します。

次のいずれかの方法を使用して、このタイムアウトを設定します。

- `ldm add-vdisk` コマンドを使用します。

```
ldm add-vdisk timeout=seconds disk-name volume-name@service-name domain-name
```

- `ldm set-vdisk` コマンドを使用します。

```
ldm set-vdisk timeout=seconds disk-name domain-name
```

- 次の行を、ゲストドメインの `/etc/system` ファイルに追加します。

```
set vdc:vdc_timeout=seconds
```

`/etc/system` プロパティ値の正しい作成または更新については、[400 ページの「`/etc/system` ファイルのプロパティ値の更新」](#)を参照してください。

注記 - この調整可能なオプションを設定すると、`ldm CLI` を使用して設定されたタイムアウトが上書きされます。また、この調整可能なオプションはゲストドメインのすべての仮想ディスクのタイムアウトを設定します。

タイムアウトは秒単位で指定します。タイムアウトを `0` に設定すると、タイムアウトは無効になり、サービスドメインの停止中は入出力がブロックされます (デフォルトの設定および動作)。

仮想ディスクおよび SCSI

物理 SCSI ディスクまたは LUN をフルディスクとしてエクスポートする場合、対応する仮想ディスクは、ユーザー SCSI コマンドインタフェース `uscsi` および多重ホストディスク制御操作 `mhd` をサポートします。バックエンドとしてファイルまたはボリュームを含む仮想ディスクなど、その他の仮想ディスクでは、これらのインタフェースはサポートされません。

注記 - `mpgroups` と SCSI 予約を一緒に使用することはできません。

そのため、SCSI コマンド (Solaris Volume Manager `metaset`、Oracle Solaris Cluster `shared devices` など) を使用するアプリケーションまたは製品機能は、バックエンドとして物理 SCSI ディスクを含む仮想ディスクのみを使用するゲストドメインで使用できます。

注記 - SCSI 操作は、仮想ディスクバックエンドとして使用される物理 SCSI ディスクまたは LUN を管理するサービスドメインによって効果的に実行されます。特に、サービスドメインは SCSI の予約を行います。このため、サービスドメインおよびゲストドメインで動作するアプリケーションは、同じ物理 SCSI ディスクに対して SCSI コマンドを発行するべきではありません。そうすると、ディスクが予期しない状態になる可能性があります。

仮想ディスクおよび format コマンド

`format` コマンドは、ドメイン上に存在するすべての仮想ディスクを認識します。ただし、1 つのスライスディスクとしてエクスポートされた仮想ディスクの場合、`format` コマンドでは、仮想ディスクのパーティションテーブルを変更できません。`label` などのコマンドは失敗しますが、書き込もうとするディスクラベルが仮想ディスクにすでに関連付けられているラベルに類似している場合は除きます。

バックエンドが SCSI ディスクである仮想ディスクでは、すべての `format(1M)` サブコマンドがサポートされています。バックエンドが SCSI ディスクでない仮想ディスクでは、一部の `format(1M)` サブコマンド (`repair`、`defect` など) がサポートされていません。この場合、`format(1M)` の動作は、Integrated Drive Electronics (IDE) ディスクの動作に類似しています。

仮想ディスクと ZFS の使用

このセクションでは、ゲストドメインにエクスポートされる仮想ディスクバックエンドを格納するために ZFS (Zettabyte File System) を使用方法について説明します。ZFS は、仮想ディスクバックエンドを作成および管理するための便利で強力なソリューションです。ZFS では、次の操作を行うことができます。

- ZFS ボリュームまたは ZFS ファイルにディスクイメージを格納する
- ディスクイメージのバックアップにスナップショットを使用する
- ディスクイメージの複製と、追加ドメインのプロビジョニングにクローンを使用する

ZFS の使用の詳細については、『[Oracle Solaris ZFS Administration Guide](#)』を参照してください。

次の説明および例で示す primary ドメインは、ディスクイメージが格納されるサービスドメインでもあります。

サービスドメインでの ZFS プールの構成

ディスクイメージを格納するには、まずサービスドメインに ZFS ストレージプールを作成します。たとえば、次のコマンドでは、primary ドメインにディスク c1t50d0 が格納された ZFS ストレージプール ldmpool が作成されます。

```
primary# zpool create ldmpool c1t50d0
```

ZFS を使用したディスクイメージの格納

次のコマンドは、ゲストドメイン ldg1 にディスクイメージを作成します。このゲストドメイン用に ZFS ファイルシステムを作成し、このゲストドメインのすべてのディスクイメージをそのファイルシステムに格納します。

```
primary# zfs create ldmpool/ldg1
```

ディスクイメージは、ZFS ボリュームまたは ZFS ファイルに格納できます。ZFS ボリュームは、サイズにかかわらず、zfs create -v コマンドを使用すると迅速に作成できます。一方、ZFS ファイルは、mkfile コマンドを使用して作成する必要があります。このコマンドの完了まで少し時間がかかることがあります。特に、作成するファイルが非常に大きいときに時間がかかり、多くはディスクイメージの作成時に該当します。

ZFS ボリュームと ZFS ファイルはいずれも、スナップショットや複製など、ZFS 機能の利点を利用できますが、ZFS ボリュームは擬似デバイス、ZFS ファイルは通常のファイルです。

OS がインストールされる仮想ディスクにディスクイメージを使用する場合、OS のインストール要件に合う十分な大きなディスクイメージが必要になります。このサイズは、OS のバージョンおよび実行されるインストールの種類によって異なります。Oracle Solaris OS をインストールする場合、20G バイトのディスクサイズを使用して、あらゆるタイプのあらゆるバージョンの Oracle Solaris OS のインストールに対応することができます。

ZFS によるディスクイメージの格納例

次の例に、ZFS ボリュームまたは ZFS ファイルを使用してディスクイメージを格納する方法を示します。ZFS ボリュームまたは ZFS ファイルをエクスポートする構文は同じですが、バックエンドへのパスは異なります。

ゲストドメインが起動すると、ZFS ボリュームまたは ZFS ファイルは、Oracle Solaris OS のインストールが可能な仮想ディスクとして表示されます。

例 10-2 ZFS ボリュームを使用したディスクイメージの格納

まず、ZFS ボリュームに 20G バイトのイメージを作成します。

```
primary# zfs create -V 20gb ldmpool/ldg1/disk0
```

次に、ZFS ボリュームを仮想ディスクとしてエクスポートします。

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldmpool/ldg1/disk0 ldg1_disk0@primary-vds0
```

ldg1 ゲストドメインに ZFS ボリュームを割り当てます。

```
primary# ldm add-vdisk disk0 ldg1_disk0@primary-vds0 ldg1
```

例 10-3 ZFS ファイルを使用したディスクイメージの格納

まず、ZFS ボリュームに 20G バイトのディスクイメージを作成し、ZFS ファイルを作成します。

```
primary# zfs create ldmpool/ldg1/disk0
primary# mkfile 20g /ldmpool/ldg1/disk0/file
```

次に、ZFS ファイルを仮想ディスクとしてエクスポートします。

```
primary# ldm add-vdsdev /ldmpool/ldg1/disk0/file ldg1_dis0@primary-vds0
```

ldg1 ゲストドメインに ZFS ファイルを割り当てます。

```
primary# ldm add-vdisk disk0 ldg1_disk0@primary-vds0 ldg1
```

ディスクイメージのスナップショットの作成

ディスクイメージが ZFS ボリュームまたは ZFS ファイルに格納されている場合は、ZFS snapshot コマンドを使用して、このディスクイメージのスナップショットを作成できます。

ディスクイメージに現在格納されているデータの一貫性を確保するため、ディスクイメージのスナップショットを作成する前に、ゲストドメインでそのディスクが現在使用されていないことを確認してください。次のいずれかの方法で、ディスクがゲストドメインで使用されていないことを確認できます。

- ゲストドメインを停止し、バインドを解除します。これはもっとも安全な対処方法であり、ゲストドメインのブートディスクとして使用されているディスクイメージのスナップショットを作成する場合に実行可能な唯一の方法です。
- ゲストドメインで使用されていて、スナップショットの対象になるディスクのスライスをアンマウントし、ゲストドメインで使用中のスライスがない状態にすることもできます。

この例では、ZFS レイアウトのため、ディスクイメージの格納場所が ZFS ボリュームまたは ZFS ファイルのどちらであっても、ディスクイメージのスナップショットを作成するコマンドは同じです。

例 10-4 ディスクイメージのスナップショットの作成

この例では、ldg1 ドメインに作成されたディスクイメージのスナップショットを作成します。

```
primary# zfs snapshot ldmpool/ldg1/disk0@version_1
```

複製を使用して新規ドメインをプロビジョニングする

ディスクイメージのスナップショットを作成したら、ZFS clone コマンドを使用してこのディスクイメージを複製できます。そのあと、複製されたイメージを別のドメインに割り当てることができます。ブートディスクイメージを複製することによって、新規ゲストドメイン用のブートディスクが迅速に作成され、Oracle Solaris OS インストールプロセス全体を実行する必要はなくなります。

たとえば、作成された disk0 がドメイン ldg1 のブートディスクである場合、次の手順を実行してこのディスクを複製し、ドメイン ldg2 のブートディスクを作成します。

```
primary# zfs create ldmpool/ldg2
primary# zfs clone ldmpool/ldg1/disk0@version_1 ldmpool/ldg2/disk0
```

ldmpool/ldg2/disk0 は、仮想ディスクとしてエクスポートして、新規の ldg2 ドメインに割り当てることができます。ドメイン ldg2 は、OS のインストールプロセスを実行しなくても、この仮想ディスクから直接ブートすることができます。

ブートディスクイメージの複製

ブートディスクを複製した場合、新しいイメージは元のブートディスクと全く同一であり、イメージの複製前にブートディスクに格納されていたホスト名、IP アドレス、マウントされているファイルシステムテーブル、システム構成、チューニングなどの情報が含まれています。

マウントされているファイルシステムテーブルは、元のブートディスクイメージ上と複製されたディスクイメージ上で同じであるため、複製されたディスクイメージは、元のドメインの場合と同じ順序で新規ドメインに割り当てる必要があります。たとえば、ブートディスクイメージが元のドメインの 1 番めのディスクとして割り当てられていた場合は、複製されたディスクイメージを新規ドメインの 1 番めのディスクとして割り当てる必要があります。このようにしない場合、新規ドメインはブートできなくなります。

元のドメインが静的 IP アドレスで構成されていた場合、複製されたイメージを使用する新規ドメインは、同じ IP アドレスで始まります。その場合は、Oracle Solaris 11 `sysconfig unconfigure` コマンドまたは Oracle Solaris 10 `sys-unconfig` コマンドを使用すると、新規ドメインのネットワーク構成を変更できます。この問題を回避するために、未構成のシステムのディスクイメージのスナップショットを作成することもできます。

元のドメインが動的ホスト構成プロトコル (Dynamic Host Configuration Protocol, DHCP) で構成されていた場合は、複製されたイメージを使用する新規ドメインも、DHCP を使用します。この場合、新規ドメインのブート時に、IP アドレスとそのネットワーク構成を自動的に受け取るため、新規ドメインのネットワーク構成を変更する必要はありません。

注記 - ドメインのホスト ID はブートディスクには格納されませんが、ドメインの作成時に Logical Domains Manager によって割り当てられます。このため、ディスクイメージを複製した場合、その新規ドメインは元のドメインのホスト ID を保持しません。

▼ 未構成システムのディスクイメージのスナップショットを作成する方法

1. 元のドメインをバインドし、起動します。
2. システムを構成解除します。

■ Oracle Solaris 11 OS: `sysconfig unconfigure` コマンドを実行します。

■ Oracle Solaris 10 OS: `sys-unconfig` コマンドを実行します。

この操作が完了すると、ドメインは停止します。

3. ドメインを停止し、バインドを解除します。ドメインをリブートしないでください。
4. ドメインのブートディスクイメージのスナップショットを作成します。

たとえば、次のように表示されます。

```
primary# zfs snapshot ldmpool/ldg1/disk0@unconfigured
```

この時点でのスナップショットは、未構成システムのブートディスクイメージです。

5. このイメージを複製して新規ドメインを作成します。このドメインの最初のブート時に、システムを構成するように求められます。

Oracle VM Server for SPARC 環境でのボリュームマネージャーの使用

このセクションでは、Oracle VM Server for SPARC 環境でのボリュームマネージャーの使用法について説明します。

ボリュームマネージャーによる仮想ディスクの使用

ZFS、Solaris Volume Manager、または Veritas Volume Manager (VxVM) ボリュームは、サービสดメインからゲストドメインに仮想ディスクとしてエクスポートできます。ボリュームは、1 つのスライスディスク (slice オプションが `ldm add-vdsdev` コマンドで指定されている場合) またはフルディスクのいずれかとしてエクスポートできます。

注記 - このセクションの残りの部分では、例として Solaris Volume Manager ボリュームを使用します。ただし、説明は ZFS および VxVM ボリュームにも適用されます。

次の例は、ボリュームを 1 つのスライスディスクとしてエクスポートする方法を示しています。

ゲストドメインの仮想ディスク (たとえば `/dev/dsk/c0d2s0`) は関連付けられたボリューム (たとえば `/dev/md/dsk/d0`) に直接割り当てられ、ゲストドメインからの仮想ディスクに格納されたデータは、メタデータを追加せずに関連付けられたボリュームに直接格納されます。そのためゲ

ストドメインからの仮想ディスクに格納されたデータは、関連付けられたボリュームを介してサービスドメインから直接アクセスすることもできます。

例

- Solaris Volume Manager ボリューム `d0` が primary ドメインから domain1 にエクスポートされる場合、domain1 の構成にはいくつかの追加の手順が必要になります。

```
primary# metainit d0 3 1 c2t70d0s6 1 c2t80d0s6 1 c2t90d0s6
primary# ldm add-vdsdev options=slice /dev/md/dsk/d0 vol3@primary-vds0
primary# ldm add-vdisk vdisk3 vol3@primary-vds0 domain1
```

- domain1 がバインドされて起動されると、エクスポートされたボリュームが `/dev/dsk/c0d2s0` のように表示され、そのボリュームが使用可能になります。

```
domain1# newfs /dev/rdisk/c0d2s0
domain1# mount /dev/dsk/c0d2s0 /mnt
domain1# echo test-domain1 > /mnt/file
```

- domain1 が停止してバインドが解除されると、domain1 から仮想ディスクに格納されたデータは、Solaris Volume Manager ボリューム `d0` を介して primary ドメインから直接アクセスできます。

```
primary# mount /dev/md/dsk/d0 /mnt
primary# cat /mnt/file
test-domain1
```

Solaris ボリュームマネージャーによる仮想ディスクの使用

RAID またはミラー Solaris Volume Manager ボリュームが別のドメインで仮想ディスクとして使用される場合は、排他 (`excl`) オプションを設定せずにエクスポートする必要があります。このようにしないと、Solaris Volume Manager ボリュームのいずれかのコンポーネントで障害が発生したときに、`metareplace` コマンドまたはホットスワップを使用した Solaris Volume Manager ボリュームの復旧が開始されません。`metastat` コマンドはそのボリュームを再同期化中と判断しますが、再同期化は進行していません。

たとえば、`/dev/md/dsk/d0` は `excl` オプションを使用して別のドメインに仮想ディスクとしてエクスポートされた RAID Solaris Volume Manager ボリュームであり、`d0` にはいくつかのホットスワップデバイスが構成されているとします。`d0` のコンポーネントに障害が発生すると、Solaris Volume Manager は障害の発生したコンポーネントをホットスワップに交換して、

Solaris Volume Manager ボリュームとの再同期化を行います。ただし、再同期化は開始されません。ボリュームは再同期化中として報告されますが、再同期化は進行していません。

```
primary# metastat d0
d0: RAID
    State: Resyncing
    Hot spare pool: hsp000
    Interlace: 32 blocks
    Size: 20097600 blocks (9.6 GB)
Original device:
    Size: 20100992 blocks (9.6 GB)
Device                               Start Block  Dbase  State Reloc
c2t2d0s1                             330        No    Okay   Yes
c4t12d0s1                             330        No    Okay   Yes
/dev/dsk/c10t600C0FF0000000000015153295A4B100d0s1 330        No    Resyncing Yes
```

このような状況で再同期化を完了するには、Solaris Volume Manager ボリュームを仮想ディスクとして使用しているドメインを停止して、バインドを解除する必要があります。そのあと、`metasync` コマンドを使用して、Solaris Volume Manager ボリュームを再同期化できます。

```
# metasync d0
```

VxVM のインストール時の仮想ディスクの使用

システムに VxVM がインストールされていて、仮想ディスクとしてエクスポートする物理ディスクまたはパーティションで Veritas Dynamic Multipathing (DMP) が有効な場合は、`excl` オプション (デフォルトではない) を設定せずにそのディスクまたはパーティションをエクスポートする必要があります。そうしない場合、このようなディスクを使用するドメインをバインドする間に `/var/adm/messages` にエラーが出力されます。

```
vd_setup_vd(): ldi_open_by_name(/dev/dsk/c4t12d0s2) = errno 16
vds_add_vd(): Failed to add vdisk ID 0
```

`vxdisk list` 出力のマルチパス化情報を調べると、Veritas DMP が有効であるかどうかを確認できます。たとえば、次のように表示されます。

```
# vxdisk list Disk_3
Device:      Disk_3
devicetag:   Disk_3
type:        auto
info:        format=none
flags:       online ready private autoconfig invalid
pubpaths:    block=/dev/vx/dmp/Disk_3s2 char=/dev/vx/rdmp/Disk_3s2
guid:        -
udid:        SEAGATE%5FST336753LSUN36G%5FDISKS%5F3032333948303144304E0000
```

```

site:      -
Multipathing information:
numpaths:  1
c4t12d0s2  state=enabled

```

また、`excl` オプションを設定して仮想ディスクとしてエクスポートするディスクまたはスライスで Veritas DMP が有効になっている場合は、`vxddmpadm` コマンドを使用して DMP を無効にすることもできます。たとえば、次のように表示されます。

```
# vxddmpadm -f disable path=/dev/dsk/c4t12d0s2
```

仮想ディスクでのボリュームマネージャーの使用

このセクションでは、仮想ディスクでのボリュームマネージャーの使用法について説明します。

仮想ディスクと ZFS の使用

仮想ディスクは ZFS とともに使用できます。ZFS ストレージプール (`zpool`) は、この `zpool` の一部であるすべてのストレージデバイスを認識する任意のドメインにインポートできます。ドメインが、これらのすべてのデバイスを仮想デバイスまたは実デバイスのどちらで認識するかは関係ありません。

仮想ディスクでの Solaris ボリュームマネージャーの使用

Solaris Volume Manager ローカルディスクセットでは、すべての仮想ディスクを使用できます。たとえば、仮想ディスクは、ローカルディスクセットの Solaris Volume Manager メタデバイス状態データベース `metadb` の格納、またはローカルディスクセットでの Solaris Volume Manager ボリュームの作成に使用できます。

バックエンドが SCSI ディスクである仮想ディスクは、Solaris Volume Manager 共有ディスクセット `metaset` で使用できます。バックエンドが SCSI ディスクでない仮想ディスクは、Solaris Volume Manager 共有ディスクセットに追加できません。バックエンドが SCSI ディスクでない仮想ディスクを Solaris Volume Manager 共有ディスクセットに追加しようとすると、次のようなエラーが表示されて失敗します。

```
# metaset -s test -a c2d2
```

```
metaset: domain1: test: failed to reserve any drives
```

仮想ディスクでの VxVM の使用

ゲストドメインでの VxVM サポートについては、Symantec 社の VxVM ドキュメントを参照してください。

仮想ディスクに関する問題

次のセクションでは、仮想ディスクを使用しているときに発生する可能性のある問題について説明します。

特定の条件で、ゲストドメインの Solaris Volume Manager 構成またはメタデバイスが失われることがある

サービสดメインが Oracle Solaris 10 1/13 OS より前のバージョンの Oracle Solaris 10 OS で動作しており、ゲストドメインに仮想ディスクとして物理ディスクスライスをエクスポートしている場合、このゲストドメインではその仮想ディスクが不適切なデバイス ID で表示されます。その後、このサービสดメインを Oracle Solaris 10 1/13 OS にアップグレードすると、このゲストドメインでは、仮想ディスクとしてエクスポートされた物理ディスクスライスはデバイス ID がない状態で表示されます。

仮想ディスクのデバイス ID を削除すると、仮想ディスクのデバイス ID を参照しようとするアプリケーションで問題が発生する可能性があります。特に、Solaris Volume Manager はその構成を検索したり、そのメタデバイスにアクセスしたりできなくなる可能性があります。

回避方法: サービสดメインを Oracle Solaris 10 1/13 OS にアップグレードしたあとで、ゲストドメインがその Solaris Volume Manager 構成またはメタデバイスを検出できない場合は、次の手順を実行します。

▼ ゲストドメインの Solaris Volume Manager 構成またはメタデバイスの検索方法

1. ゲストドメインをブートします。

2. 次の行を `/kernel/drv/md.conf` ファイルに追加することによって、Solaris Volume Manager の `devid` 機能を無効にします。

```
md_devid_destroy=1;
md_keep_repl_state=1;
```

3. ゲストドメインをリブートします。

ドメインがブートされたあと、Solaris Volume Manager 構成とメタデバイスを検出できるようになります。

4. Solaris Volume Manager 構成をチェックして、それが正しいことを確認します。

5. 手順 2 で追加した 2 行を `/kernel/drv/md.conf` ファイルから削除することによって、Solaris Volume Manager の `devid` 機能をふたたび有効にします。

6. ゲストドメインをリブートします。

リブート時に次のようなメッセージが表示されます。

```
NOTICE: mddb: unable to get devid for 'vdc', 0x10
```

これらのメッセージは正常で、問題を報告するものではありません。

Oracle Solaris ブートディスクの互換性

歴史的に見て、Oracle Solaris OS は SMI VTOC ディスクラベルで構成されたブートディスクにインストールされてきました。Oracle Solaris 11.1 OS 以降は、デフォルトで拡張ファームウェアインタフェース (EFI) GUID パーティションテーブル (GPT) ディスクラベルを使って構成されるブートディスクに OS がインストールされます。ファームウェアが EFI をサポートしていない場合は、代わりに SMI VTOC ディスクラベルを使ってディスクが構成されます。この状況は、少なくともシステムファームウェアバージョン 8.4.0 を実行している SPARC T4 サーバー、少なくともシステムファームウェアバージョン 9.1.0 を実行している SPARC T5 サーバー、SPARC M5 サーバー、および SPARC M6 サーバー、少なくともシステムファームウェアバージョン 9.4.3 を実行している SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバー、少なくともシステムファームウェアバージョン XCP2230 を実行している Fujitsu M10 サーバーにのみ適用されます。

次のサーバーは、EFI GPT ディスクラベルの付いたディスクからブートできません。

- 使用されるシステムファームウェアのバージョンに関係なく、UltraSPARC T2、UltraSPARC T2 Plus、および SPARC T3 サーバー

- バージョン 8.4.0 より前のシステムファームウェアが実行されている SPARC T4 サーバー
- 9.1.0 より前のシステムファームウェアバージョンを実行している SPARC T5 サーバー、SPARC M5 サーバー、および SPARC M6 サーバー
- 9.4.3 より前のシステムファームウェアバージョンを実行している SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバー

そのため、最新の SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、または SPARC M7 シリーズサーバー上で作成された Oracle Solaris 11.1 ブートディスクを、それより古いサーバー、またはそれより古いファームウェアを実行しているサーバー上で使用することはできません。

この制限により、コールド移行またはライブ移行のどちらかを使って新しいサーバーから古いサーバーにドメインを移すことができなくなります。また、この制限により、古いサーバーで EFI GPT ブートディスクイメージを使用することもできなくなります。

Oracle Solaris 11.1 ブートディスクが、使用しているサーバーやそのファームウェアと互換性があるかどうかを判断するには、Oracle Solaris 11.1 OS が SMI VTOC ディスクラベルで構成されたディスクにインストールされていることを確認します。

古いファームウェアが動作しているシステムとの下位互換性を維持するには、次のいずれかの手順を使用します。そうしないと、ブートディスクはデフォルトで EFI GPT ディスクラベルを使用します。これらの手順は、Oracle Solaris 11.1 OS が、少なくともシステムファームウェアバージョン 8.4.0 を実行している SPARC T4 サーバー上、少なくともシステムファームウェアバージョン 9.1.0 を実行している SPARC T5 サーバー、SPARC M5 サーバー、または SPARC M6 サーバー上、および少なくともシステムファームウェアバージョン 9.4.3 を実行している SPARC T7 シリーズサーバーまたは SPARC M7 シリーズサーバー上の SMI VTOC ディスクラベルの付いたブートディスクにインストールされていることを確認する方法を示しています。

- **解決方法 1:** ファームウェアが EFI をサポートしていることを報告しないように gpt プロパティを削除します。

1. OpenBoot PROM プロンプトから、自動ブートを無効にし、インストールされるシステムをリセットします。

```
ok setenv auto-boot? false
ok reset-all
```

システムがリセットされると、ok プロンプトに戻ります。

2. /packages/disk-label ディレクトリに変更し、gpt プロパティを削除します。


```
ok cd /packages/disk-label
ok " gpt" delete-property
```

3. Oracle Solaris 11.1 OS のインストールを開始します。

たとえば、ネットワークインストールを実行します。

```
ok boot net - install
```

- **解決方法 2:** `format -e` コマンドを使用して、Oracle Solaris 11.1 OS がインストールされるディスクに SMI VTOC ラベルを書き込みます。

1. ディスクに SMI VTOC ラベルを書き込みます。

たとえば、`label` オプションを選択して、SMI ラベルを指定します。

```
# format -e c1d0
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
```

2. ディスク全体をカバーするスライス 0 とスライス 2 でディスクを構成します。

このディスクにはそのほかにパーティションはありません。たとえば、次のように表示されます。

```
format> partition

partition> print
Current partition table (unnamed):
Total disk cylinders available: 14087 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 14086	136.71GB	(14087/0/0) 286698624
1	unassigned	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 14086	136.71GB	(14087/0/0) 286698624
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

3. SMI VTOC ディスクラベルを再度書き込みます。

```
partition> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 0
Ready to label disk, continue? y
```

4. Oracle Solaris OS をブートディスクのスライス 0 にインストールするように Oracle Solaris Automatic Installer (AI) を構成します。

AI マニフェストの <disk> の抜粋を次のように変更します。

```
<target>
  <disk whole_disk="true">
    <disk_keyword key="boot_disk"/>
    <slice name="0" in_zpool="rpool"/>
  </disk>
[...]
```

5. Oracle Solaris 11.1 OS のインストールを実行します。

◆◆◆ 第 11 章

仮想 SCSI ホストバスアダプタの使用

この章では、Oracle VM Server for SPARC ソフトウェアで仮想 SCSI ホストバスアダプタ (HBA) を使用する方法について説明します。

この章では、次の項目について説明します。

- [219 ページの「仮想 SCSI ホストバスアダプタの概要」](#)
- [222 ページの「仮想 SCSI HBA の動作モデル」](#)
- [223 ページの「仮想 SCSI HBA の識別子とデバイス名」](#)
- [224 ページの「仮想 SCSI HBA の管理」](#)
- [229 ページの「ゲストドメイン内の仮想 LUN の見え方」](#)
- [229 ページの「仮想 SCSI HBA と仮想 SAN の構成」](#)
- [230 ページの「仮想 SCSI HBA マルチパスの構成」](#)
- [234 ページの「SCSI デバイスからのブート」](#)
- [235 ページの「仮想 LUN のインストール」](#)
- [236 ページの「仮想 SCSI HBA のタイムアウト」](#)
- [236 ページの「仮想 SCSI HBA および SCSI」](#)
- [237 ページの「ゲストドメインでの断片化の激しい I/O バッファのサポート」](#)

仮想 SCSI ホストバスアダプタの概要

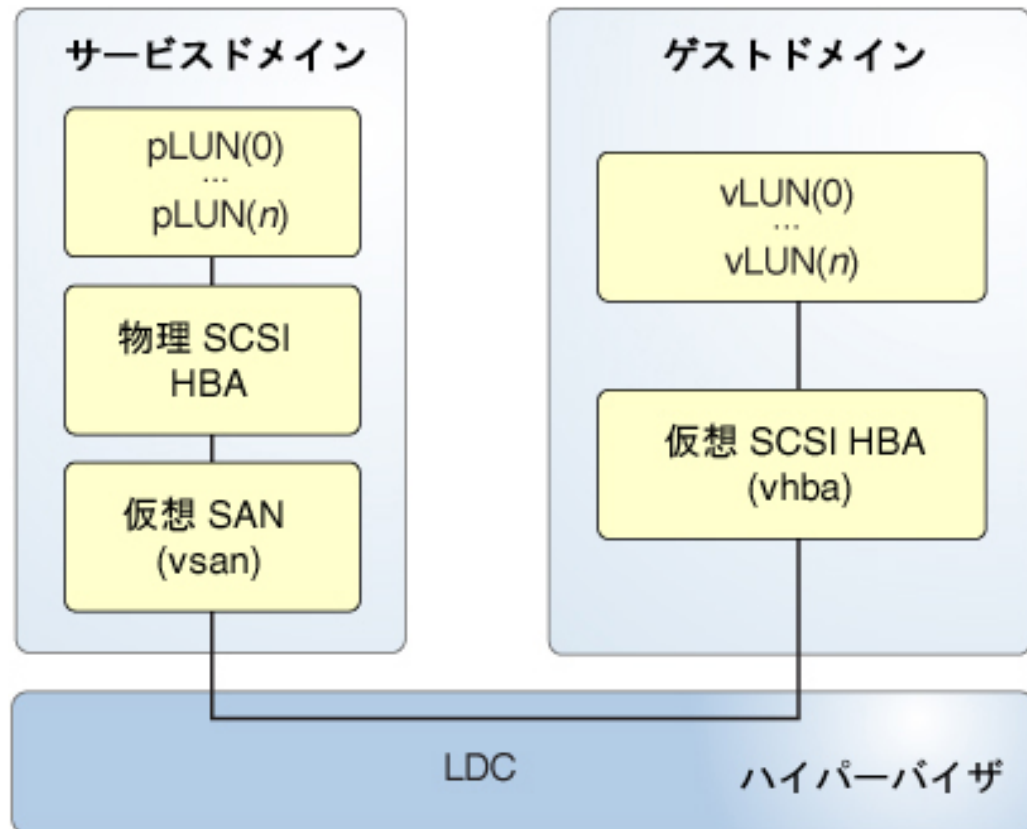
仮想 SCSI ホストバスアダプタ (HBA) は、ゲストドメイン内の仮想 HBA とサービスドメイン内の仮想ストレージエリアネットワーク (SAN) という 2 つのコンポーネントで構成されます。仮想 HBA と仮想 SAN インスタンスは連携して、ゲストドメイン内で実行される SCSI ターゲットドライバのための SCSI HBA インタフェースを実装します。vSAN サービスは、SCSI I/O 要求をサービスドメイン内で実行される物理 SCSI HBA ドライバに渡す vsan ドライバによって実装されます。vhba ドライバは、ハイパーバイザで管理される論理ドメインチャネル (LDC) を使用して I/O 要求を vsan に送信します。

vHBA インスタンスは、特定の vSAN インスタンスから到達可能なすべての SCSI デバイスへのアクセスを提供します。vHBA は、ディスク、CD、DVD、テープなどのすべての SCSI デバイスタイプを認識できます。一連の到達可能な SCSI デバイスは、仮想 SAN に関連付けられた物理 HBA ドライバから現在認識可能な一連の物理 SCSI デバイスに基づいて変化します。特定の vHBA から認識可能な SCSI デバイスの識別情報や数は実行時まで不明ですが、これは物理 HBA ドライバでも同様に起こります。

vHBA にはその子デバイスとして仮想 LUN (vLUN) が存在し、これらは物理 LUN と同じ方法で動作します。たとえば、vHBA インスタンスとその vLUN でネイティブな Oracle Solaris マルチパスソリューション (MPxIO) を使用できます。vLUN のデバイスパスは、完全な `cXtYdZsN` 表記法である `/dev/[r]disk/cXtYdZsN` を使用します。このデバイス名の `tY` の部分は、SCSI ターゲットデバイスを示します。

仮想 SAN と仮想 SCSI HBA を構成したら、`format` コマンドを使用して、OpenBoot プロンプトからの仮想 LUN のブートや、すべての仮想 LUN の表示などの操作を実行できます。

図 11-1 Oracle VM Server for SPARC での仮想 SCSI HBA



仮想 SAN がサービスドメイン内に存在し、`vsan` カーネルモジュールによって実装されるのに対して、仮想 SCSI HBA はゲストドメイン内に存在し、`vhba` モジュールによって実装されます。仮想 SAN が特定の物理 SCSI HBA イニシエータポートに関連付けられているのに対して、仮想 SCSI HBA は特定の仮想 SAN に関連付けられています。

`vhba` モジュールは、SCSA 準拠のすべての SCSI ターゲットドライバからの I/O 要求を受信するために、SCSA 準拠のインタフェースをエクスポートします。`vhba` モジュールは I/O 要求を、LDC 経由でサービスドメインに送信される仮想 I/O プロトコルメッセージに変換します。

`vsan` モジュールは、`vhba` によって送信された仮想 I/O メッセージを I/O 要求に変換します。これらの要求は、SCSA 準拠の物理 SCSI HBA ドライバに送信されます。`vsan` モジュールは、

I/O ペイロードとステータスを LDC 経由で `vhba` に返します。最後に、`vhba` は、この I/O 応答を I/O 要求の発信元に転送します。

仮想 SCSI HBA の動作モデル

仮想 SCSI HBA を使用するための動作モデルは、Logical Domains Manager では仮想 SCSI HBA と仮想 SAN インスタンスしか認識できないため、ほかのタイプの Oracle VM Server for SPARC 仮想デバイスの場合とは質的に異なります。ゲストドメイン内に表示される仮想 LUN や、サービスドメイン内に表示される物理 LUN は、実行時に検出されるまで不明です。仮想 LUN や物理 LUN は、関連付けられた LDC 接続がリセットされたときには暗黙的に、また `ldm rescan-vhba` コマンドを使用した場合は明示的に検出されます。

`ldm` コマンドを使用して仮想ディスクを明示的に指定しても、ゲストドメイン内の仮想 LUN は、その識別情報をサービスドメイン内の関連付けられた物理 LUN の識別情報から取得します。[ldm\(1M\)](#) マニュアルページを参照してください。

たとえば、物理 LUN と仮想 LUN は、次のデバイスパス内の太字で示されているテキストを共有します。

■ サービスドメイン内の物理 LUN:

```
/pci@0/pci@0/pci@8/pci@0/pci@2/SUNW,qlc@0/fp@0,0/ssd@w216000c0ff8089d5,0
```

■ ゲストドメイン内の仮想 LUN:

```
/virtual-devices@100/channel-devices@200/scsi@1/iport@0/disk@w216000c0ff8089d5,0
```

注記 - このゲストドメインのデバイスパスは、ゲストドメインで MPxIO が無効になっている場合にのみ存在します。MPxIO が有効になっている場合、`scsi_vhci` モジュールはゲストドメイン内にデバイスパスを作成しますが、その構文は異なります。

仮想 LUN のデバイスパス内の `scsi@1` コンポーネントは、この仮想 LUN がメンバーになっている仮想 SCSI HBA インスタンスを示すことに注意してください。

仮想 SCSI HBA の一連の仮想 LUN は実行時にサービスドメインから取得されるため、仮想 LUN を明示的にゲストドメインに追加したり削除したりすることはできません。代わりに、ゲストドメインの仮想 LUN のメンバーシップを変更できるように、基本となる物理 LUN を追加または削除する必要があります。ドメインのリブートやドメインの移行などのイベントによって、ゲストドメインの仮想 LUN のメンバーシップが変更される可能性があります。この変更は、仮想 SCSI HBA の LDC 接続がリセットされると常に、仮想 LUN が自動的に再検出されるために発生します。仮想 LUN の基本となる物理 LUN が将来の検出で見つからない場合、その

仮想 LUN は使用不可としてマークされ、その仮想 LUN にアクセスすると次のようなエラーが返されます。

```
WARNING: .../scsi@1/iport@0/disk@w216000c0ff8089d5,0 (sd6): ... Command failed to complete...
Device is gone
```

仮想 SCSI HBA インスタンスは `vhba` ドライバによって管理されますが、仮想 LUN は、基本となる物理 LUN のデバイスタイプに基づいて SCSI ターゲットドライバによって管理されます。次の出力は、`vhba` ドライバが仮想 SCSI HBA インスタンスを管理していること、および `sd` SCSI ディスクドライバが仮想 LUN を管理していることを確認します。

```
# prtconf -a -D /dev/dsk/c2t216000C0FF8089D5d0
SUNW,SPARC-Enterprise-T5220 (driver name: rootnex)
  virtual-devices, instance #0 (driver name: vnex)
    channel-devices, instance #0 (driver name: cnex)
      scsi, instance #0 (driver name: vhba)
        iport, instance #3 (driver name: vhba)
          disk, instance #30 (driver name: sd)
```

仮想 SCSI HBA の識別子とデバイス名

`ldm add-vhba` コマンドを使用して仮想 SCSI HBA をドメインに追加する場合は、`id` プロパティを設定することによって、そのデバイス番号を指定できます。

```
ldm add-vhba [id=vHBA-ID] vHBA-name vSAN-name domain-name
```

ドメインの各仮想 SCSI HBA には、そのドメインがバインドされるときに割り当てられる一意のデバイス番号があります。仮想 SCSI HBA が (`id` プロパティを 10 進数値に設定することにより) 明示的なデバイス番号で追加された場合は、指定されたデバイス番号が使用されます。デバイス番号を指定しなかった場合、使用可能なもっとも小さいデバイス番号が自動的に割り当てられます。その場合、割り当てられるデバイス番号は、その仮想 SCSI HBA がドメインに追加された方法によって異なります。ドメインがバインドされると、仮想 SCSI HBA に最終的に割り当てられたデバイス番号が `ldm list-bindings` および `ldm list -o hba` コマンドの出力に表示されます。

`ldm list-bindings`、`ldm list -o hba`、および `ldm add-vhba id=id` コマンドはすべて、`id` プロパティ値を 10 進数値として表示および指定します。Oracle Solaris OS は、仮想 SCSI HBA の `id` 値を 16 進数として表示します。

次の例は、`vhba@0` デバイスが `gdom` ドメイン上の `vh1` 仮想 SCSI HBA のデバイス名であることを示しています。

```
primary# ldm list -o hba gdom
```

```
NAME
```

```
gdom
```

```
VHBA
```

```
NAME
```

```
VSAN
```

```
DEVICE
```

```
TOUT
```

```
SERVER
```

```
vh1
```

```
vs1
```

```
vhba@0
```

```
0
```

```
svcdom
```



注意 - 仮想 SCSI HBA にデバイス番号が明示的に割り当てられていない場合は、ドメインがバインド解除され、あとで再バインドされると、そのデバイス番号が変更されることがあります。その場合は、ドメインで実行されている OS によって割り当てられたデバイス名も変更され、それによりシステムの既存の構成が破壊される可能性があります。これは、たとえば、ドメインの構成から仮想 SCSI HBA が削除された場合に発生することがあります。

仮想 SCSI HBA の管理

このセクションでは、次のタスクについて説明します。

- [224 ページの「物理 SCSI HBA に関する情報の取得」](#)
- [225 ページの「仮想ストレージエリアネットワークの作成」](#)
- [226 ページの「仮想 SCSI ホストバスアダプタの作成」](#)
- [227 ページの「仮想 SCSI HBA の存在の確認」](#)
- [227 ページの「仮想 SCSI HBA のタイムアウトオプションの設定」](#)
- [228 ページの「仮想 SCSI ホストバスアダプタの削除」](#)
- [228 ページの「仮想ストレージエリアネットワークの削除」](#)
- [228 ページの「LUN の追加または削除」](#)

このセクションに示されているコマンドの詳細は、[ldm\(1M\)](#) のマニュアルページを参照してください。

物理 SCSI HBA に関する情報の取得

仮想 SCSI HBA を構成する前に、サービスドメインに接続されている物理 SCSI HBA に関する情報を取得する必要があります。I/O ドメインでの HBA カードの構成の詳細は、[第5章「I/O ドメインの構成」](#)を参照してください。

注記 - primary ドメイン内に少なくとも Oracle Solaris 11.3 OS がインストールされている場合は、サービスドメインを制御ドメインにできます。

`ldm list-hba` コマンドは、指定されたアクティブなドメインの物理 SCSI HBA イニシエータポートを一覧表示します。論理ドメインの SCSI HBA イニシエータポートを識別したら、`ldm add-vsan` コマンド行で特定のイニシエータポートを指定して仮想 SAN を作成できます。

```
ldm list-hba [-d] [-l] [-p] [-t] domain-name
```

次の例は、`svcdom` サービスドメインに接続されている SCSI HBA のイニシエータポートを示しています。`-l` オプションは、詳細情報を表示します。

```
primary# ldm list-hba -l svcdom
NAME                                     VSAN
----                                     -
/SYS/MB/SASHBA0/HBA0/PORT1
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@1]
/SYS/MB/SASHBA0/HBA0/PORT2
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@2]
/SYS/MB/SASHBA0/HBA0/PORT4
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@4]
/SYS/MB/SASHBA0/HBA0/PORT8
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@8]
/SYS/MB/PCIE1/HBA0/PORT0,0
[/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0/fp@0,0]
/SYS/MB/PCIE1/HBA0,1/PORT0,0
[/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0,1/fp@0,0]
```

イニシエータポートに対して表示されると予測した LUN が `ldm list-hba` の出力に表示されていない場合は、参照されたイニシエータポートの参照されたサービスドメインでマルチパスが無効になっていることを確認してください。[『Managing SAN Devices and Multipathing in Oracle Solaris 11.3』](#)を参照してください。

バインドされたサービスドメインに仮想 SAN を追加すると、ゲストドメインへの接続が確立されない場合があります。次の現象が現れることがあります。

- 仮想 SCSI HBA からのディスクが `format` の出力に表示されない
- 仮想 LUN からブートすると、LDC エラーが表示される

この問題を解決するには、`/etc/driver/drv/vsan.conf` ファイルに `ddi-no-autodetach="yes"` という行を追加してから、サービスドメインをリブートします。

仮想ストレージエリアネットワークの作成

物理 SCSI HBA のイニシエータポートを取得したあと、サービスドメイン上に仮想ストレージエリアネットワーク (SAN) を作成する必要があります。仮想 SAN は、指定された SCSI HBA イニシエータポートから到達可能なすべての SCSI デバイスを管理します。

```
ldm add-vsan [-q] iport-path vSAN-name domain-name
```

vSAN 名は、指定されたドメイン名ではなく、制御ドメインに対して一意です。このドメイン名は、SCSI HBA イニシエータポートが構成されているドメインを識別します。同じイニシエータポートパスを参照する複数の仮想 SAN を作成できます。

`ldm add-vsan` コマンドを使用して、イニシエータポートパスを 1 つまたは複数の仮想 SAN に構成できます。この構成により、制御ドメイン内の複数のサービスドメインが同じイニシエータポートパスを使用できるようになります。

注記 - サービスドメイン上で Oracle Solaris 11.3 OS が実行されている場合、`ldm add-vsan` コマンドは、イニシエータポートパスが有効なデバイスパスであることを検証します。`ldm add-vsan` コマンドを実行したときに、指定されたサービスドメインがアクティブでない場合、そのサービスドメインでは指定されたイニシエータポートパスを検証できません。イニシエータポートパスが、サービスドメインの一部であるインストールされている物理 SCSI HBA イニシエータポートに対応していない場合は、そのサービスドメインがアクティブになったあとに、サービスドメインのシステムログに警告メッセージが書き込まれます。

この例では、`svcdom` サービスドメイン上の `/SYS/MB/PCIE1/HBA0,1/PORT0,0` イニシエータポートを仮想 SAN に関連付けます。仮想 SAN の名前を選択できます。この例では、`port0` が仮想 SAN の名前です。

```
primary# ldm add-vsan /SYS/MB/PCIE1/HBA0,1/PORT0,0 port0 svcdom
/SYS/MB/PCIE1/HBA0,1/PORT0,0 resolved to device:
/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0,1/fp@0,0
```

仮想 SCSI ホストバスアダプタの作成

仮想 SAN が定義されたら、`ldm add-vhba` コマンドを使用して、ゲストドメイン内に仮想 SCSI HBA を作成できます。仮想 SCSI HBA は、I/O 要求を仮想 SAN 内の物理 SCSI デバイスに送信します。

```
ldm add-vhba [id=vHBA-ID] vHBA-name vSAN-name domain-name
```

この例では、`port0` 仮想 SAN と通信する `gdom` ゲストドメイン上に `port0_vhba` 仮想 SCSI HBA を作成します。

```
primary# ldm add-vhba port0_vhba port0 gdom
```

仮想 SCSI HBA の存在の確認

サービスドメインおよびゲストドメイン上に新しく作成された仮想 SCSI HBA と仮想 SAN デバイスの存在を確認するには、`ldm list` コマンドを使用します。

```
ldm list -o san,hba [domain-name ...]
```

この例では、仮想 SAN を含むサービスドメインが `svcdom` であり、仮想 SCSI HBA を含むゲストドメインが `gdom` です。`gdom` ドメインはまだバインドされていないため、この例では仮想 HBA の識別子が割り当てられていないことに注意してください。

```
primary# ldm list -o san,hba svcdom gdom
NAME
svcdom

VSAN
  NAME          TYPE  DEVICE IPORT
  port0         VSAN  [/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0,1/fp@0,0]

-----
NAME
gdom

VHBA
  NAME          VSAN          DEVICE TOUT SERVER
  port0_vhba    port0          0      svcdom
```

仮想 SCSI HBA のタイムアウトオプションの設定

`ldm set-vhba` コマンドでは、指定された論理ドメイン上の仮想 SCSI HBA のタイムアウト値を指定できます。`timeout` プロパティは、指定された仮想 SCSI HBA インスタンスが仮想 SAN との LDC 接続を確立できないことを宣言する前に待機する時間 (秒単位) を指定します。[236 ページの「仮想 SCSI HBA のタイムアウト」](#)を参照してください。

のデフォルトのタイムアウト値 0 を指定すると、`vhba` は、仮想 SAN との LDC 接続を作成するために無期限に待機します。

```
ldm set-vhba [timeout=seconds] vHBA-name domain-name
```

この例では、`gdom` ゲストドメイン上の `port0_vhba` 仮想 SCSI HBA に対して 90 セクションのタイムアウトを設定します。

```
primary# ldm set-vhba timeout=90 port0_vhba gdom
```

仮想 SCSI ホストバスアダプタの削除

`ldm remove-vhba` コマンドを使用すると、指定されたゲストドメインから仮想 SCSI HBA を削除できます。

削除を試行する前に、その仮想 SCSI HBA が OS とアプリケーションのどちらからもアクティブに使用されていないことを確認してください。その仮想 SCSI HBA が使用中の場合、`ldm remove-vhba` コマンドは失敗します。

```
ldm remove-vhba vHBA-name domain-name
```

この例では、`gdom` ゲストドメインから `port0_vhba` 仮想 SCSI HBA を削除します。

```
primary# ldm remove-vhba port0_vhba gdom
```

仮想ストレージエリアネットワークの削除

`ldm remove-vsan` コマンドを使用すると、仮想 SAN を削除できます。

まず、その仮想 SAN に関連付けられた仮想 SCSI HBA を削除します。次に、`ldm remove-vsan` コマンドを使用して仮想 SAN を削除します。

```
ldm remove-vsan vSAN-name
```

この例では、`port0` 仮想 SAN を削除します。

```
primary# ldm remove-vsan port0
```

LUN の追加または削除

仮想 SCSI HBA から直接仮想 LUN を追加または削除することはできません。まず物理 LUN を追加または削除し、次に `ldm rescan-vhba` コマンドを実行して、仮想 SCSI HBA および仮想 SAN によって認識される一連の SCSI デバイスを同期する必要があります。物理 LUN を追加または削除するためのコマンドは、仮想 SAN に関連付けられたイニシエータポートのトポロジによって異なります。たとえば、イニシエータポートが物理 SAN と通信する場合は、SAN 管理コマンドを使用して SAN 要素に LUN を追加するか、または SAN 要素から LUN を削除する必要があります。

```
ldm rescan-vhba vHBA-name domain-name
```

たとえば、次のコマンドは、gdom ドメイン上の port0_vhba 仮想 SCSI HBA の SCSI デバイスを同期します。

```
primary# ldm rescan-vhba port0_vhba gdom
```

ゲストドメイン内の仮想 LUN の見え方

仮想 SCSI HBA に関連付けられた (1 つまたは複数の) 仮想 LUN は、あたかも物理 LUN であるかのように動作します。

たとえば、SCSI ディスクを表す仮想 LUN は、ゲストドメインでは /dev/[r]dsk の下の通常のディスクとして見えます。基本となる関連付けられた物理 LUN のタイプは disk であるため、仮想 LUN は format コマンドの出力に表示されます。仮想 LUN のデバイスパスは、prtpticl や prtconf などのほかのコマンドで操作できます。

仮想 LUN が CD または DVD を表す場合、そのデバイスパスは /dev/[r]dsk で定義されています。仮想 LUN がテープデバイスを表す場合、そのデバイスパスは /dev/rmt で定義されています。仮想 LUN を操作できるコマンドについては、『[Managing Devices in Oracle Solaris 11.3](#)』を参照してください。

仮想 SCSI HBA と仮想 SAN の構成

仮想 SCSI HBA と仮想 SAN の構成は非常に柔軟です。ldm add-vsant コマンドによって使用される物理 SCSI HBA イニシエータポートは、SCSI をサポートするあらゆるタイプのバス (ファイバチャネル、SAS、SATA など) を駆動できます。仮想 SCSI HBA と仮想 SAN は、同じドメイン内で実行できます。仮想 SAN はまた、ldm add-io コマンドを使用して物理 SCSI HBA カードをサービスドメインに追加したあと、I/O ドメイン内でも実行できます。

仮想 SAN は、概念的には物理 SAN に関連付けられますが、その関連付けは必要ありません。サーバーの 1 つまたは複数のローカルディスクで構成される仮想 SAN を作成できます。たとえば、一部のシステムには、次の ldm list-hba の出力で SASHBA として示されているマザーボードの SAS HBA から到達可能なディスクが存在します。

```
primary# ldm list-hba -d primary
NAME MY_VSAN
```

```
-----
```

```
/SYS/MB/SASHBA0/HBA0/PORT4
c5t5000CCA0564DEF39d0s0
/SYS/MB/SASHBA0/HBA0/PORT1
c3t5000CCA0564F1A7Dd0s0
/SYS/MB/SASHBA0/HBA0/PORT2
c4t5000CCA0564F6B89d0s0
/SYS/MB/SASHBA0/HBA0/PORT8
c6t5000CCA0564FCF6Dd0s0
```

サーバーのローカルディスクをカプセル化する仮想 SAN を定義する場合は、primary ドメインのブート元のディスクの仮想 LUN を誤って作成してしまわないように、必ず次の `zpool` コマンドを使用してください。たとえば、次の `zpool` コマンドは、root rpool が `/SYS/MB/SASHBA0/HBA0/PORT2` イニシエータポートから到達可能なディスク `c4t5000CCA0564F6B89d0` 上にマウントされることを確認します。

```
# zpool iostat -v
```

pool	capacity		operations		bandwidth	
	alloc	free	read	write	read	write
-----	-----	-----	-----	-----	-----	-----
rpool	25.0G	531G	0	10	257	81.7K
c4t5000CCA0564F6B89d0	25.0G	531G	0	10	257	81.7K
-----	-----	-----	-----	-----	-----	-----

仮想 SAN を構成する場合は、LUN 0 を持つ SCSI ターゲットデバイスの物理 LUN のみがゲストドメインに表示されることに注意してください。この制約は、ターゲットの LUN 0 が SCSI REPORT LUNS コマンドに応答することを要求する Oracle Solaris OS 実装によって課されます。

仮想 SCSI HBA マルチパスの構成

仮想 SCSI HBA サブシステムは、ネイティブな Oracle Solaris マルチパス実装 (MPxIO) を利用することによってゲストドメイン内のマルチパスをサポートします。詳細は、『[Managing SAN Devices and Multipathing in Oracle Solaris 11.3](#)』を参照してください。

ネイティブなマルチパスと同様に、特定のバックエンド SCSI デバイスに 1 つ以上のパスでアクセスできます。仮想 SCSI HBA サブシステムの場合は、各パスが 1 つの仮想 LUN に関連付けられます。`scsi_vhci` モジュールは、関連付けられた `mpathadm` 管理コマンドに渡された引数に基づいて I/O 要求を一連の仮想 LUN に送信するネイティブなマルチパス動作を実装します。詳細は、[scsi_vhci\(7D\)](#) および [mpathadm\(1M\)](#) のマニュアルページを参照してください。

マルチパスを構成するには、ゲストドメインから同じバックエンドデバイスへの 2 つ以上の個別のパスを構成する必要があります。マルチパスが 1 つの構成されたパスでも動作することに

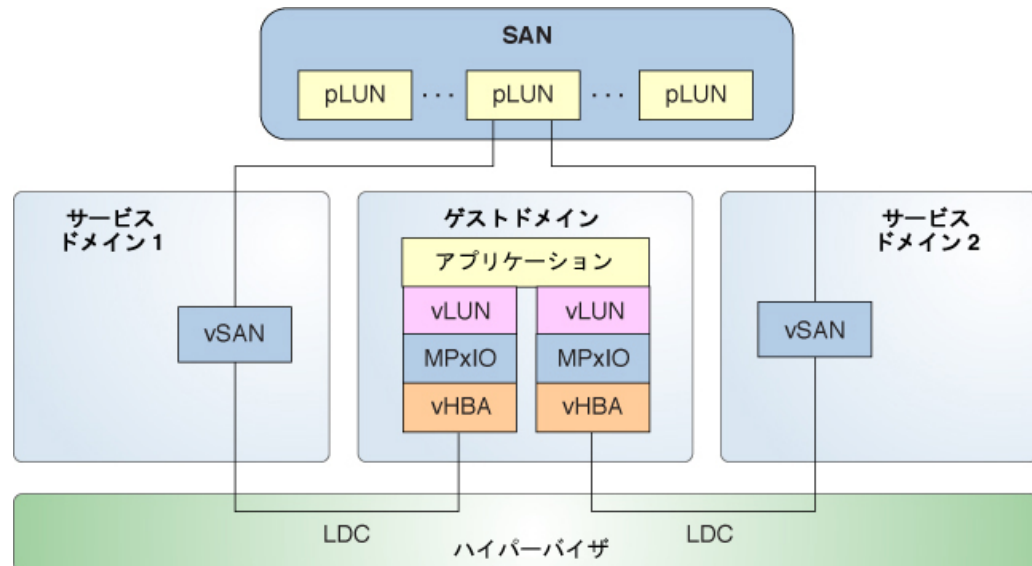
注意してください。ただし、予測される構成には、個別のサービスドメイン上に存在する個別の物理 SCSI HBA イニシエータポート経由で I/O 要求を送信する 2 つ以上のパスが存在します。

1. バックエンドストレージへの個別のパスごとに `ldm add-vhba` コマンドと `ldm add-vsan` コマンドのペアを実行します。
2. `vhba` 仮想 HBA モジュールによって管理されるイニシエータポートに対してゲストドメイン内のネイティブなマルチパスを有効にします。

注記 - 仮想 SCSI HBA サブシステムは、サービスドメイン内のネイティブなマルチパスの有効化をサポートしていません。表示されると予測した LUN が特定のサービスドメインに対する `ldm list-hba` の出力に表示されていない場合は、それらの LUN に関連付けられた物理 SCSI HBA イニシエータポートに対して MPxIO が無効になっていることを確認してください。必要に応じて、`stmsboot` コマンドを使用してイニシエータポートの MPxIO を無効にします。

次の図は、マルチパス構成の例です。これは、MPxIO によって管理されている 2 つのパスによってアクセスされる SAN の 1 つの物理 LUN を示しています。この図に示されている構成の作成方法を説明する手順については、[232 ページの「仮想 SCSI HBA マルチパスを構成する方法」](#)を参照してください。

図 11-2 仮想 SCSI HBA マルチパスの構成



▼ 仮想 SCSI HBA マルチパスを構成する方法

この手順では、[図11-2「仮想 SCSI HBA マルチパスの構成」](#)に示されている仮想 SCSI HBA マルチパス構成を作成する方法について説明します。この例は、考えられる多くのマルチパス構成のうちの 1 つにすぎません。

1. 物理 SCSI HBA が割り当てられた I/O ドメインを作成します。

[第5章「I/O ドメインの構成」](#)を参照してください。

2. 最初のサービスドメインから最初のイニシエータポートパスの仮想 SAN をエクスポートします。

```
ldm add-vsan vSAN-path1 vSAN-name domain-name
```

vSAN-path1 は、この SAN への最初のイニシエータポートパスです。

たとえば、次のコマンドは、svcdom1 ドメインから vsan-mpxio1 仮想 SAN をエクスポートします。

```
primary# ldm add-vsan /SYS/MB/RISER0/PCIE1/HBA0/PORT0 vsan-mpxio1 svcdom1
```

3. 2 番目のサービスドメインから 2 番目のイニシエータポートパスの仮想 SAN をエクスポートします。

```
ldm add-vsan vSAN-path2 vSAN-name domain-name
```

vSAN-path2 は、この SAN への 2 番目のイニシエータポートパスです。

たとえば、次のコマンドは、svcdom2 ドメインから vsan-mpxio2 仮想 SAN をエクスポートします。

```
primary# ldm add-vsan /SYS/MB/RISER0/PCIE3/HBA0/PORT0 vsan-mpxio2 svcdom2
```

4. 仮想 SCSI HBA をゲストドメインにエクスポートします。

```
ldm add-vhba vHBA-name vSAN-name domain-name
```

たとえば、次のコマンドは、vhba-mpxio1 および vhba-mpxio2 仮想 SCSI HBA を gdom ドメインにエクスポートします。

```
primary# ldm add-vhba vhba-mpxio1 vsan-mpxio1 gdom
primary# ldm add-vhba vhba-mpxio2 vsan-mpxio2 gdom
```

5. ゲストドメイン上の仮想 SCSI HBA の timeout プロパティ値を指定します。


```
ldm set-vhba timeout=seconds vHBA-name domain-name
```

たとえば、次のコマンドは、gdom ドメイン上の vsan-mpxio1 および vsan-mpxio2 仮想 SCSI HBA の timeout プロパティ値を 30 に設定します。

```
primary# ldm set-vhba timeout=30 vhba-mpxio1 gdom
primary# ldm set-vhba timeout=30 vhba-mpxio2 gdom
```

6. ゲストドメインをリブートします。

▼ 仮想 SCSI HBA のマルチパスを有効にする方法

このタスクはゲストドメイン上で実行します。

1. /platform/sun4v/kernel/drv/vhba.conf ファイルを /etc/driver/drv ディレクトリにコピーします。

```
# cp /platform/sun4v/kernel/drv/vhba.conf /etc/driver/drv
```
2. /etc/driver/drv/vhba.conf ファイルを編集して mpxio-disable プロパティを no に設定することによって MPxIO を有効にします。
3. ゲストドメインをリブートします。

▼ 仮想 SCSI HBA のマルチパスを無効にする方法

このタスクはゲストドメイン上で実行します。

1. /platform/sun4v/kernel/drv/vhba.conf ファイルを /etc/driver/drv ディレクトリにコピーします。

```
# cp /platform/sun4v/kernel/drv/vhba.conf /etc/driver/drv
```
2. /etc/driver/drv/vhba.conf ファイルを編集して mpxio-disable プロパティを yes に設定することによって MPxIO を無効にします。
3. ゲストドメインをリブートします。

SCSI デバイスからのブート

次のセクションでは、SCSI デバイスからブートする方法について説明します。

- [234 ページの「仮想 LUN からのブート」](#)
- [235 ページの「SCSI DVD デバイスからのブート」](#)

仮想 LUN からのブート

関連付けられた物理 LUN が OBP でブート可能な SCSI デバイスタイプ (CD、DVD、ディスクなど) を参照している任意の仮想 LUN をブートできます。

OpenBoot PROM プロンプトで `boot` コマンドを発行する前に、ゲストドメインの仮想 SCSI HBA および関連付けられた仮想 LUN を検索するために `probe-scsi-all` コマンドを実行します。

次の注釈付きの例では、その出力の関連する部分を強調表示しています。

```
{0} ok probe-scsi-all

/virtual-devices@100/channel-devices@200/scsi@0          行 1

vHBA

TPORT-PHYS: w200200110d214900                             行 2
  LUN: 1   Disk   VLUN           2097152 Blocks, 1073 MB
  LUN: 0   Disk   VLUN           32768000 Blocks, 16 GB    行 3
```

この `probe-scsi-all` の出力例は、タイプが `disk` である 2 つの LUN を持つ 1 つの仮想 SCSI HBA インスタンス (`scsi@0`) を示しています。

特定の仮想 LUN からブートするには、`boot` コマンドに渡すデバイスパスを手動で構成します。デバイスパスの構文は次のとおりです。

```
vhba-device-path/disk@target-port,lun:slice
```

行 3 の LUN からブートするには、デバイスパスを次のように構成する必要があります。

- `target-port` の値を行 2 から取得する
- `vhba-device-path` の値を行 1 から取得する

結果として得られるデバイスパスを次に示します。

```
/virtual-devices@100/channel-devices@200/scsi@0/disk@w200200110d214900,0
```

このデバイスパスを OBP の boot コマンドに次のように渡すことができます。

```
{0} ok boot /virtual-devices@100/channel-devices@200/scsi@0/disk@w200200110d214900,0
```

SCSI DVD デバイスからのブート

SCSI デジタル多用途ディスク (DVD) ドライブからブートして、その DVD からゲストドメインをインストールできます。

次の例は、SCSI DVD デバイスが primary ドメインに接続されている仮想 SCSI HBA の構成を示しています。

```
primary# ldm list-hba -t -d primary
IPOINT                                VSAN
-----                                ----
[...]
/SYS/MB/SASHBA1/HBA0/PORT40
    init-port w50800200008f4329
    Transport Protocol SAS
    c2t3d0s0
    lun 0
    removable media 1

primary# ldm add-vsan /SYS/MB/SASHBA1/HBA0/PORT40 dvd_vsan primary
/SYS/MB/SASHBA1/HBA0/PORT40 resolved to device: /pci@400/pci@2/pci@0/pci@4/scsi@0/iport@40
primary# ldm add-vhba dvd_vhba dvd_vsan gdom
```

gdom ドメインコンソールから、SCSI デバイスをプローブして DVD からブートします。

```
{0} ok probe-scsi-all
/virtual-devices@100/channel-devices@200/scsi@0

vHBA

TPORT-PHYS: p3
    LUN: 0    Removable Read Only device    TEAC    DV-W28SS-V 1.0B

{0} ok boot /virtual-devices@100/channel-devices@200/scsi@0/disk@p3
...
```

仮想 LUN のインストール

関連付けられた物理 LUN がインストールプログラムでサポートされるタイプを持つ SCSI デバイスを参照している任意の仮想 LUN に OS をインストールできます。そのあと、指定された仮想 LUN からブートできます。

仮想 SCSI HBA のタイムアウト

デフォルトでは、仮想 SAN へのアクセスを提供するサービスドメインが使用できない場合、ゲストドメインから対応する仮想 SCSI HBA への I/O はすべてブロックされます。そのサービスドメインが動作可能になり、仮想 SAN へのサービスを復元すると、I/O は自動的に再開されます。

サービスドメインが使用できない時間が長すぎる場合は、ファイルシステムまたはアプリケーションが I/O 操作に対して、失敗してエラーを報告するよう要求することがあります。各仮想 SCSI HBA がゲストドメイン上の仮想 SCSI HBA とサービスドメイン上の仮想 SAN の間の接続を確立するための接続タイムアウト期間を設定できます。そのタイムアウト期間に達すると、サービスドメインが使用できるようになり、仮想 SCSI HBA と仮想 SAN の間の接続が再確立されないかぎり、保留中の I/O および新しい I/O 操作はすべて失敗します。

タイムアウト値を指定する必要があるその他の状況には、次のものがあります。

- MPxIO が別の構成されたパスにフェイルオーバーするようにする場合は、関係する各仮想 SCSI HBA についてタイムアウトを設定する必要があります。
- ライブ移行を実行する場合は、移行されるゲストドメイン内の各仮想 SCSI HBA について `timeout` プロパティ値を 0 に設定します。移行が完了したら、各仮想 SCSI HBA についてタイムアウトプロパティを元の設定にリセットします。

タイムアウト値を設定する方法を見つけるには、[227 ページの「仮想 SCSI HBA のタイムアウトオプションの設定」](#)を参照してください。

仮想 SCSI HBA および SCSI

`vhba` モジュールは、仮想 SAN の SCSI イニシエータポートに関連付けられた物理 SCSI HBA ドライバへの SCSI コマンドをプロキシ設定します。

ネイティブな Oracle Solaris マルチパスを実装する `scsi_vhci` ドライバは、SCSI-2 予約と SCSI-3 予約の両方のパスフェイルオーバー中の予約の永続性を処理します。`vhba` モジュールは、Oracle Solaris I/O フレームワークにプラグインし、`scsi_vhci` のサポートを利用することによって SCSI 予約をサポートします。

ゲストドメインでの断片化の激しい I/O バッファのサポート

ほかの sun4v 仮想デバイスと同様に、vhba モジュールは、ソフトウェアスタック内の上位レイヤーによって作成された I/O バッファで動作します。その I/O バッファがあまりにも多くの物理メモリのフラグメントの寄せ集めである場合、vhba モジュールは、I/O 要求の処理時に次の致命的な警告メッセージを発行します。

```
WARNING: ... ldc_mem_bind_hdl: ncookies(max, actual) = (8, 9)
```

各物理メモリアフラグメントが 1 つの cookie に関連付けられています。cookie の実際の数を cookie の最大数によってサポートできない場合、I/O 要求は失敗します。

このエラーメッセージは、必要な cookie の実際の数を示しています。このエラーを解消するには、使用する I/O バッファごとの cookie の数を指定する `/etc/system` ファイル内の `vhba_desc_ncookies` 値を実際の値以上に変更します。また、cookie の許容可能な最大数を指定する `vhba_desc_max_ncookies` プロパティの値も増やします。

`/etc/system` プロパティ値の正しい作成または更新については、[400 ページの「/etc/system ファイルのプロパティ値の更新」](#)を参照してください。

そのあと、`ldm remove-vhba` および `ldm add-vhba` コマンドを連続して実行するか、またはゲストドメインをリブートすることによって仮想 SCSI HBA の接続を再度作成します。

たとえば、`vhba_desc_max_ncookies` プロパティ値を 12 に設定するには、`/etc/system` ファイルに次の行を追加します。

```
set vhba:vhba_desc_ncookies = 12
```


◆◆◆ 第 12 章

仮想ネットワークの使用

この章では、Oracle VM Server for SPARC ソフトウェアで仮想ネットワークを使用する方法について説明します。この章の内容は次のとおりです。

- 240 ページの「仮想ネットワークの概要」
- 240 ページの「Oracle Solaris 11 ネットワークの概要」
- 243 ページの「Oracle Solaris 10 ネットワークの概要」
- 245 ページの「仮想ネットワークパフォーマンスの最大化」
- 247 ページの「仮想スイッチ」
- 248 ページの「仮想ネットワークデバイス」
- 251 ページの「ネットワークデバイス構成および統計情報の表示」
- 254 ページの「仮想ネットワークデバイスで消費される物理ネットワーク帯域幅の量の制御」
- 258 ページの「仮想デバイス識別子およびネットワークインタフェース名」
- 261 ページの「自動または手動による MAC アドレスの割り当て」
- 264 ページの「Oracle Solaris 10 を実行するドメインでのネットワークアダプタの使用」
- 265 ページの「NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成」
- 269 ページの「Oracle VM Server for SPARC 環境での IPMP の構成」
- 279 ページの「VLAN のタグ付けの使用」
- 284 ページの「プライベート VLAN の使用」
- 290 ページの「パケットスループットパフォーマンスの調整」
- 291 ページの「NIU ハイブリッド I/O の使用」
- 296 ページの「仮想スイッチでのリンクアグリゲーションの使用」
- 299 ページの「ジャンボフレームの構成」
- 304 ページの「Oracle Solaris 11 のネットワーク固有の機能の相違点」
- 306 ページの「仮想ネットワーク上の仮想 NIC の使用」

Oracle Solaris OS のネットワークは Oracle Solaris 10 OS と Oracle Solaris 11 OS の間で大幅に変更されました。考慮する問題については、[243 ページの「Oracle Solaris 10 ネットワークの概要」](#)、[240 ページの「Oracle Solaris 11 ネットワークの概要」](#)、[304 ページの「Oracle Solaris 11 のネットワーク固有の機能の相違点」](#)を参照してください。

仮想ネットワークの概要

仮想ネットワークでは、ドメインが外部の物理ネットワークを使用しないで相互に通信できます。仮想ネットワークでは、複数のドメインが同じ物理ネットワークインタフェースを使用して物理ネットワークにアクセスし、リモートシステムと通信することもできます。仮想ネットワークは、仮想ネットワークデバイスを接続できる仮想スイッチを備えることで構築します。

Oracle Solaris のネットワークは Oracle Solaris 10 OS と Oracle Solaris 11 OS で大きく違います。次の 2 つのセクションでは、各 OS のネットワークに関する概要を説明します。

注記 - Oracle Solaris 10 のネットワークは、ドメインまたはシステムで同じように動作します。Oracle Solaris 11 ネットワークについても同様です。Oracle Solaris OS のネットワークの詳細は、[Oracle Solaris 10 のドキュメント](#)および [Oracle Solaris 11.3 のドキュメント](#)を参照してください。

Oracle Solaris 10 と Oracle Solaris 11 のネットワークの機能の違いについては、[240 ページの「Oracle Solaris 11 ネットワークの概要」](#)で説明しています。

Oracle Solaris 11 ネットワークの概要

Oracle Solaris 11 OS では、多くの新しいネットワーク機能が導入されました。これらについては、[Oracle Solaris 11.3 のドキュメント](#)にある Oracle Solaris 11 ネットワークドキュメントで説明されています。

Oracle VM Server for SPARC ソフトウェアを使用する場合、次の Oracle Solaris 11 ネットワーク機能を理解することが重要です。

- すべてのネットワーク構成は、`ipadm` コマンドと `dladm` コマンドで実行します。
- 「デフォルトのバニティー名」機能は、すべての物理ネットワークアダプタに対して、`net0` などの汎用リンク名を生成します。この機能は、仮想スイッチ (`vswin`) および仮想ネットワー

クデバイス (`vnetn`) の汎用名も生成し、これらは、OS から物理ネットワークアダプタのように見えます。物理ネットワークデバイスに関連付けられている汎用リンク名を特定するには、`dladm show-phys` コマンドを使用します。

デフォルトで Oracle Solaris 11, 物理ネットワークデバイス名には汎用の「バニティー」名が使用されます。Oracle Solaris 10. で使用されていた `nxge0` などのデバイスドライバ名の代わりに、`net0` などの汎用名が使用されます。

次のコマンドは、`nxge0` などのドライバ名の代わりに、汎用名 `net0` を指定して、`primary` ドメインの仮想スイッチを作成します。

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

- Oracle Solaris 11 OS は仮想ネットワークインタフェースカード (VNIC) を使用して、内部仮想ネットワークを作成します。

VNIC は物理ネットワークデバイスから作成し、ゾーンに割り当てることができる物理ネットワークデバイスの仮想インスタンスです。

■

Oracle VM Server for SPARC ソフトウェアの構成時に、Oracle Solaris 11 `DefaultFixed` ネットワーク構成プロファイル (NCP) を使用します。

Oracle Solaris 11 ドメインの場合、`DefaultFixed` NCP を使用します。このプロファイルは、インストール中またはインストール後に有効にできます。Oracle Solaris 11 のインストール時に、手動ネットワーク構成を選択します。

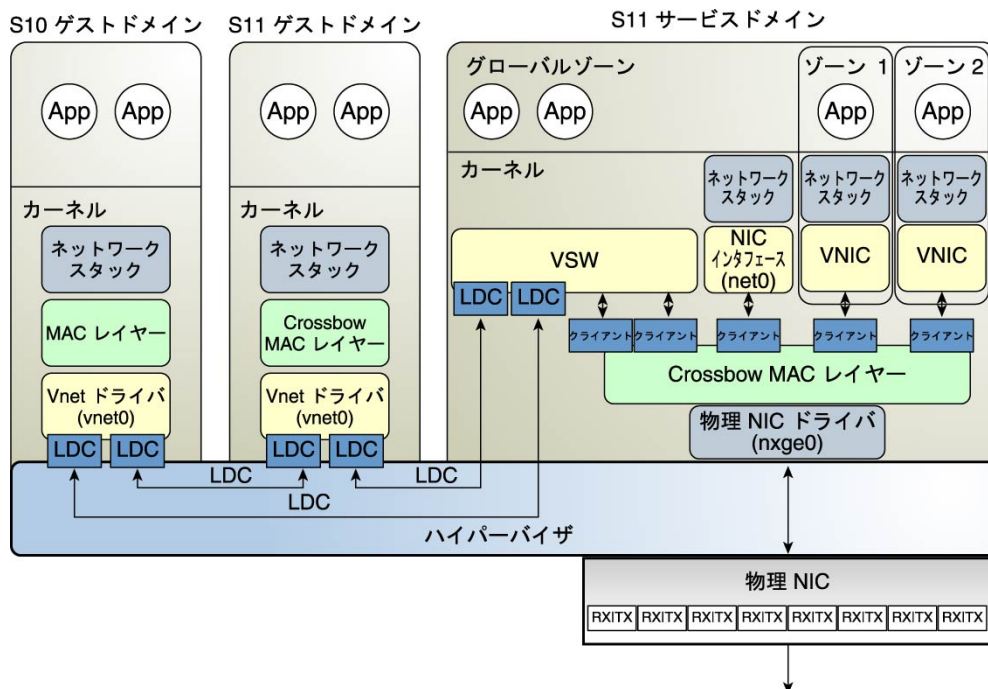
- プライマリネットワークインタフェースを仮想スイッチ (`vsw`) インタフェースで置換しないでください。サービスドメインは、既存のプライマリネットワークインタフェースを使用して、同じ仮想スイッチに接続されている仮想ネットワークデバイスを含むゲストドメインと通信できます。
- 仮想スイッチに物理ネットワークアダプタの MAC アドレスを使用すると、プライマリネットワークインタフェースと競合するため、仮想スイッチに物理ネットワークアダプタの MAC アドレスを使用しないでください。

注記 - このリリースでは、`dladm` または `ipadm` コマンドを使用して Oracle Solaris 11 システム上にデータリンクおよびネットワークインタフェースを構成するために、DefaultFixed NCP を使用します。

`netadm list` コマンドを使用して、DefaultFixed NCP が有効になっていることを確認します。『[Oracle Solaris Administration: Network Interfaces and Network Virtualization](#)』の第 7 章「[Using Datalink and Interface Configuration Commands on Profiles](#)」を参照してください。

次の図は、Oracle Solaris 10 OS を実行するゲストドメインが Oracle Solaris 11 サービスドメインと完全な互換性があることを示しています。唯一の違いは、Oracle Solaris 11 OS で追加または拡張された機能です。

図 12-1 Oracle Solaris 11 OS の Oracle VM Server for SPARC ネットワークの概要



図は、`nxge0` や `vnet0` などのネットワークデバイス名を、Oracle Solaris 11 ドメインで `netn` などの汎用リンク名で表現できることを示しています。また、次の点にも注意してください。

- サービスドメインの仮想スイッチはゲストドメインに接続され、ゲストドメインが互いに通信できます。
- 仮想スイッチは物理ネットワークデバイス `nxge0` にも接続され、ゲストドメインが物理ネットワークと通信できます。

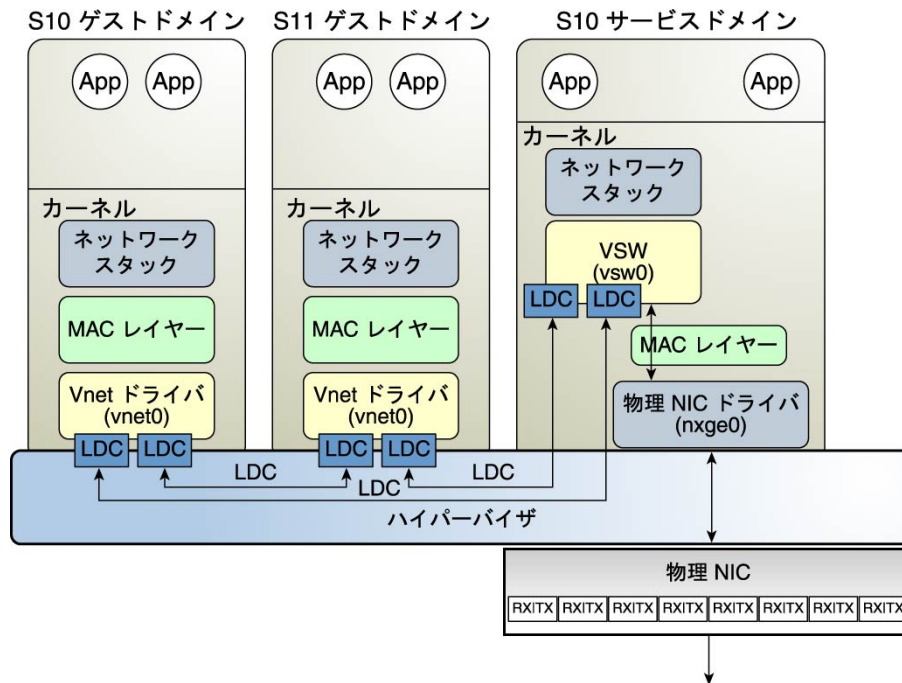
仮想スイッチにより、ゲストドメインはサービスドメインネットワークインタフェース `net0` に加え、`nxge0` と同じ物理ネットワークデバイス上の VNIC とも通信できます。これには、ゲストドメインと Oracle Solaris 11 サービスドメインの間の接続が含まれます。仮想スイッチ自体 (`vswi` デバイス) をネットワークデバイスとして構成しないでください。この機能は Oracle Solaris 11 では非推奨になり、サポートされなくなっています。
- Oracle Solaris 10 ゲストドメイン内の仮想ネットワークデバイス `vnet0` は、`ifconfig` コマンドを使用してネットワークインタフェースとして構成できます。
- Oracle Solaris 11 ゲストドメイン内の仮想ネットワークデバイス `vnet0` は、`net0` などの汎用リンク名で表示されることがあります。`ipadm` コマンドを使ってネットワークインタフェースとして構成できます。

仮想スイッチは、通常の物理ネットワークスイッチと同様に動作し、それが接続されているさまざまなシステム間で、ネットワークパケットをスイッチングします。システムは、ゲストドメイン、サービスドメイン、物理ネットワークなどです。

Oracle Solaris 10 ネットワークの概要

次の図は、Oracle Solaris 11 OS を実行するゲストドメインが Oracle Solaris 10 サービスドメインと完全な互換性があることを示しています。唯一の違いは、Oracle Solaris 11 OS で追加または拡張された機能です。

図 12-2 Oracle Solaris 10 OS の Oracle VM Server for SPARC ネットワークの概要



前の図は、Oracle Solaris 10 OS のみに適用される `nxge0`、`vsw0`、`vnet0` などのインタフェース名を示しています。また、次の点にも注意してください。

- サービスドメインの仮想スイッチはゲストドメインに接続され、ゲストドメインが互いに通信できます。
- 仮想スイッチは物理ネットワークインタフェース `nxge0` にも接続され、ゲストドメインが物理ネットワークと通信できます。
- 仮想スイッチネットワークインタフェース `vsw0` がサービスドメイン内で作成され、2 つのゲストドメインはサービスドメインと通信できます。
- サービスドメイン内の仮想スイッチネットワークインタフェース `vsw0` は、Oracle Solaris 10 `ifconfig` コマンドを使用して構成できます。
- Oracle Solaris 10 ゲストドメイン内の仮想ネットワークデバイス `vnet0` は、`ifconfig` コマンドを使用してネットワークインタフェースとして構成できます。

- Oracle Solaris 11 ゲストドメイン内の仮想ネットワークデバイス `vnet0` は、`net0` などの汎用リンク名で表示されることがあります。`ipadm` コマンドを使ってネットワークインタフェースとして構成できます。

仮想スイッチは、通常の物理ネットワークスイッチと同様に機能し、接続されているゲストドメイン、サービスドメイン、物理ネットワークなど異なるシステム間のネットワークパケットをスイッチングします。`vsw` ドライバは、仮想スイッチをネットワークインタフェースとして構成することができるネットワークデバイス機能を備えています。

仮想ネットワークパフォーマンスの最大化

このセクションの説明に従ってプラットフォームとドメインを構成すれば、ゲストと外部ネットワークとの通信およびゲスト間の通信の転送速度を速くすることができます。仮想ネットワークスタックでは、ジャンボフレームを使用する必要なく、高い TCP パフォーマンスを実現する LSO (Large Segment Offload) がサポートされています。

ハードウェアおよびソフトウェアの要件

ドメインのネットワークパフォーマンスを最大にするには、次の要件を満たします。

- **ハードウェア要件。**これらのパフォーマンス向上は、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、および SPARC M7 シリーズサーバーでのみ使用できます。
- **システムファームウェア要件。**これらの SPARC システムは、最新のシステムファームウェアを実行する必要があります。[『Oracle VM Server for SPARC 3.3 インストールガイド』の「完全に認定されたシステムファームウェアバージョン」](#)を参照してください。
- **Oracle Solaris OS 要件。**サービスドメインおよびゲストドメインで次の Oracle Solaris OS バージョンが実行されていることを確認してください。

注記 - 完全に認定された Oracle Solaris OS バージョンを実行すると、新機能へのアクセスが提供されます。[『Oracle VM Server for SPARC 3.3 インストールガイド』の「完全に認定された Oracle Solaris OS バージョン」](#)を参照してください。

- サービスドメイン。150031-03 パッチが適用された Oracle Solaris 11.1 SRU 9 OS または Oracle Solaris 10 OS 以上。
- ゲストドメイン。150031-03 パッチが適用された Oracle Solaris 11.1 SRU 9 OS または Oracle Solaris 10 OS 以上。
- CPU およびメモリー要件。サービスドメインおよびゲストドメインに十分な CPU およびメモリーリソースが割り当てられていることを確認します。
- サービスドメイン。サービスドメインはゲストドメインのデータプロキシとして機能するため、サービスドメインに 2 CPU 以上のコアおよび 4G バイト以上のメモリーを割り当てます。
- ゲストドメイン。10G ビット/秒以上のパフォーマンスが実現されるように、各ゲストドメインを構成します。各ゲストドメインに 2 CPU 以上のコアおよび 4G バイト以上のメモリーを割り当てます。

仮想ネットワークのパフォーマンスを最大にするためのドメインの構成

旧バージョンの Oracle VM Server for SPARC および Oracle Solaris OS では、ジャンボフレームを構成することでネットワークパフォーマンスを改善できました。この構成は必須ではなくなったため、別の理由で必要な場合を除いて、サービスドメインおよびゲストドメインでは標準の MTU 値 1500 を使用することが最適な方法です。

ネットワークパフォーマンスの向上を実現するには、サービスドメインおよびゲストドメインの `extended-mapin-space` プロパティを `on` に設定します。これはデフォルトの動作です。

```
primary# ldm set-domain extended-mapin-space=on domain-name
```

`extended-mapin-space` プロパティ値を確認するには、次のコマンドを実行します。

```
primary# ldm list -l domain-name |grep extended-mapin  
extended-mapin-space=on
```

注記 - `extended-mapin-space` プロパティ値を変更すると、`primary` ドメインで遅延再構成がトリガーされます。このような状況では、`primary` ドメインのリブートが必要です。また、このプロパティ値を変更する前に、まずゲストドメインを停止する必要があります。

仮想スイッチ

仮想スイッチ (vsw) とは、サービスドメインで動作し、仮想スイッチドライバによって管理されるコンポーネントのことです。仮想スイッチを複数のゲストドメインに接続すると、これらのドメイン間のネットワーク通信を可能にできます。また、仮想スイッチが物理ネットワークインタフェースにも関連付けられている場合は、物理ネットワークインタフェースを介して、ゲストドメインと物理ネットワークの間のネットワーク通信が許可されます。Oracle Solaris 10 サービスドメインで実行されている場合、仮想スイッチもネットワークインタフェース `vswn` を持ち、その仮想スイッチに接続されているほかのドメインとの通信をサービスドメインに許可します。仮想スイッチは、通常のネットワークインタフェースのように使用でき、Oracle Solaris 10 `ifconfig` コマンドで構成できます。

ドメインへ仮想ネットワークデバイスを割り当てると、仮想スイッチを提供するドメインで暗黙的な依存関係が作成されます。`ldm list-dependencies` コマンドを使用すると、これらの依存関係を表示する、またはこの仮想スイッチに依存するドメインを表示することができます。[424 ページの「ドメイン I/O の依存関係の一覧表示」](#)を参照してください。

Oracle Solaris 11 サービスドメインでは、仮想スイッチを通常のネットワークインタフェースとして使用できません。仮想スイッチが物理ネットワークインタフェースに接続されている場合、サービスドメインとの通信はこの物理インタフェースを使用することによって可能です。物理インタフェースなしで構成されている場合は、`etherstub` を VNIC と接続されているネットワークデバイス (`net-dev`) を使用して、サービスドメインとの通信を有効にできます。

仮想スイッチのバックエンドデバイスとして使用するネットワークデバイスを判定するには、`dladm show-phys` 出力で物理ネットワークデバイスを検索するか、`ldm list-netdev` コマンドを使用して論理ドメインのネットワークデバイスを一覧表示します。

注記 - Oracle Solaris 10 サービスドメインに仮想スイッチを追加する場合、そのネットワークインタフェースは作成されません。このため、デフォルトでは、サービスドメインは仮想スイッチに接続されたゲストドメインと通信できません。ゲストドメインとサービスドメインの間のネットワーク通信を有効にするには、関連付けられた仮想スイッチのネットワークインタフェースを作成し、サービスドメイン内で構成する必要があります。手順については、[47 ページの「Oracle Solaris 10 サービスドメインとその他のドメインの間のネットワークの有効化」](#)を参照してください。

この状況は、Oracle Solaris 10 OS でのみ発生し、Oracle Solaris 11 OS では発生しません。

仮想スイッチのドメインへの追加、仮想スイッチへのオプションの設定、仮想スイッチの削除には、それぞれ、`ldm add-vsw` コマンド、`ldm set-vsw` コマンド、および `ldm rm-vsw` コマンドを使用できます。[ldm\(1M\)](#) マニュアルページを参照してください。

NIC またはアグリゲーションの VLAN タグ付きインスタンスに仮想スイッチを作成する場合、`ldm add-vsw` または `ldm set-vsw` コマンドを使用する際に、`net-dev` プロパティの値として、NIC (`nxge0`)、アグリゲーション (`aggr3`)、またはバニティー名 (`net0`) を指定する必要があります。

注記 - Oracle Solaris 11.2 SRU 1 OS 以降では、`ldm set-vsw` コマンドを使用して `net-dev` プロパティ値を動的に更新できます。以前の Oracle Solaris OS リリースでは、`ldm set-vsw` コマンドを使用して `primary` ドメインの `net-dev` プロパティ値を更新すると、`primary` ドメインは遅延再構成に入ります。

IP-over-InfiniBand (IPoIB) ネットワーク・デバイス上に仮想スイッチを追加することはできません。`ldm add-vsw` コマンドと `ldm add-vnet` コマンドは成功したように見えますが、これらのデバイスは InfiniBand トランスポート層によって IP パケットをトランスポートするので、データは流れません。仮想スイッチはトランスポート層として Ethernet のみをサポートします。

次のコマンドは、`net0` という名前の物理ネットワークアダプタ上に仮想スイッチを作成します。

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

次のサンプルは、`ldm list-netdev -b svcdom` コマンドを使用して、`svcdom` サービスドメインの有効な仮想スイッチバックエンドデバイスのみを表示します。

```
primary# ldm list-netdev -b svcdom
DOMAIN
svcdom
```

NAME	CLASS	MEDIA	STATE	SPEED	OVER	LOC
net0	PHYS	ETHER	up	10000	ixgbe0	/SYS/MB/RISER1/PCIE
net1	PHYS	ETHER	unknown	0	ixgbe1	/SYS/MB/RISER1/PCIE4
net2	ESTUB	ETHER	unknown	0	--	--
net3	ESTUB	ETHER	unknown	0	--	--
ldoms-estub.vsw0	ESTUB	ETHER	unknown	0	--	--

仮想ネットワークデバイス

仮想ネットワークデバイスとは、仮想スイッチに接続されたドメイン内で定義されている仮想デバイスのことです。仮想ネットワークデバイスは、仮想ネットワークドライバによって管理され、論

理ドメインチャネル (LDC) を使用するハイパーバイザを介して仮想ネットワークに接続されます。

仮想ネットワークデバイスは、`vnet $n$` という名前のネットワークインタフェースとして使用でき、通常のネットワークインタフェースと同様に使用して、Oracle Solaris 10 `ifconfig` コマンドまたは Oracle Solaris 11 `ipadm` コマンドで構成できます。

注記 - Oracle Solaris 11 の場合、デバイスには汎用名が割り当てられるため、`vnet $n$` は `net0` などの汎用名を使用します。

仮想ネットワークデバイスのドメインへの追加、既存の仮想ネットワークデバイスへのオプション設定、仮想ネットワークデバイスの削除には、それぞれ `ldm add-vnet` コマンド、`ldm set-vnet` コマンド、および `ldm rm-vnet` コマンドを使用できます。[ldm\(1M\)](#) マニュアルページを参照してください。

[図12-2「Oracle Solaris 10 OS の Oracle VM Server for SPARC ネットワークの概要」](#)と[図12-1「Oracle Solaris 11 OS の Oracle VM Server for SPARC ネットワークの概要」](#)にある、それぞれ Oracle Solaris 10 と Oracle Solaris 11 での Oracle VM Server for SPARC のネットワークに関する情報を参照してください。

Inter-Vnet LDC チャネル

デフォルトで、Logical Domains Manager は、次の方法で LDC チャネルを割り当てます。

- LDC チャネルは、仮想ネットワークデバイスと仮想スイッチデバイス間に割り当てられます。
- LCD チャネルは、同じ仮想スイッチデバイス (inter-vnet) に接続される仮想ネットワークデバイスの組み合わせごとに割り当てられます。

inter-vnet LDC チャネルは、ゲスト間に高度な通信パフォーマンスを確立するために、仮想ネットワークデバイスが直接通信するように構成されます。ただし、仮想スイッチデバイス内の仮想ネットワークデバイスの数が増えるにつれて、vnet 間通信に必要な LDC チャネルの数は 2 次関数的に増加します。

指定の仮想スイッチデバイスに接続されたすべての仮想ネットワークデバイスに対して、inter-vnet LDC チャネルの割り当てを有効または無効にすることを選択できます。この割り当てを無効にすると、数が限られている LDC チャネルの消費量を削減できます。

この割り当てを無効にすることは、次のような状況で役立ちます。

- ゲスト間の通信パフォーマンスが最優先の重要事項ではない

■ 1 台の仮想スイッチデバイスに多数の仮想ネットワークデバイスが必要である

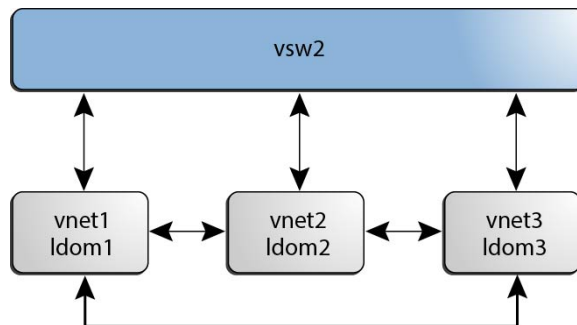
inter-vnet チャンネルを割り当てないことで、仮想 I/O デバイスをゲストドメインに追加するために、より多くの LDC チャンネルが利用可能になります。

注記 - システム内の仮想ネットワークデバイス数の増加よりも、ゲスト間のパフォーマンスの重要性が高い場合は、inter-vnet LDC チャンネルの割り当てを無効にしないでください。

ldm add-vsw および ldm set-vsw コマンドを使用すると、inter-vnet-link プロパティに on または off の値を指定できます。

次の図は、3 つの仮想ネットワークデバイスを保有する一般的な仮想スイッチを表します。inter-vnet-link プロパティが on に設定されていると、inter-vnet LDC チャンネルが割り当てられていることを意味します。vnet1 および vnet2 間でのゲスト間通信は、仮想スイッチを経由せずに直接実行されます。

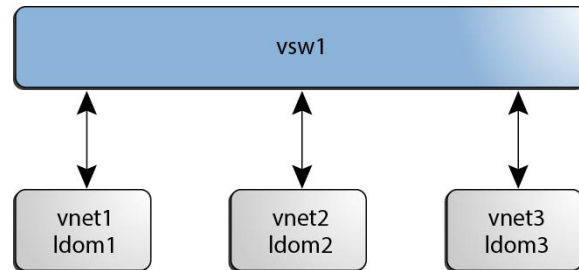
図 12-3 Inter-Vnet チャンネルを使用する仮想スイッチの構成



次の図では、同じ仮想スイッチ構成の inter-vnet-link プロパティが off に設定されています。Inter-vnet LDC チャンネルは割り当てられていません。inter-vnet-link プロパティが on に設定された場合よりも少ない LDC チャンネルが使用されます。この構成では、vnet1 および vnet2 間のゲスト間通信は vsw1 を経由する必要があります。

注記 - inter-vnet LDC チャンネルの割り当てを無効にしても、ゲスト間の通信は妨げられません。代わりに、すべてのゲスト間の通信トラフィックは、1 つのゲストドメインから別のゲストドメインへ直接移動するのではなく、仮想スイッチを経由します。

図 12-4 Inter-Vnet チャンネルを使用しない仮想スイッチの構成



ネットワークデバイス構成および統計情報の表示

ldm list-netdev コマンドと ldm list-netstat コマンドにより、システム内のネットワークデバイスに関する情報およびネットワーク統計情報をそれぞれ表示できます。その結果、所定の物理ドメイン内のネットワークデバイスおよび統計情報を一元的に表示できます。

これらのコマンドを使用するには、ゲストドメインで少なくとも Oracle Solaris 11.2 SRU 1 OS を実行する必要があります。

例 12-1 ネットワークデバイス構成情報の一覧表示

次の例は、ldg1 ドメインのネットワークデバイスの短い一覧を ldm list-netdev コマンドを使用して表示します。

```
primary# ldm list-netdev ldg1
```

```
DOMAIN
ldg1
```

NAME	CLASS	MEDIA	STATE	SPEED	OVER	LOC
net0	VNET	ETHER	up	0	--	primary-vsw0/vne t0_ldg1
net3	PHYS	ETHER	up	10000	--	/SYS/MB/RISER1/PCIE4
net4	VSW	ETHER	up	10000	--	ldg1-vsw1
net1	PHYS	ETHER	up	10000	--	/SYS/MB/RISER1/PCIE4
net5	VNET	ETHER	up	0	--	ldg1-vsw1/vnet1_ldg1
net6	VNET	ETHER	up	0	--	ldg1-vsw1/vnet2_ldg1
aggr2	AGGR	ETHER	unknown	0	net1,net3	--
ldoms-vsw0.vport3	VNIC	ETHER	unknown	0	--	ldg1-vsw1/vnet2_ldg1

```
ldoms-vsw0.vport2 VNIC  ETHER unknown 0    --    ldg1-vsw1/vnet1_ldg1
ldoms-vsw0.vport1 VNIC  ETHER unknown 0    --    ldg1-vsw1/vnet2_ldg3
ldoms-vsw0.vport0 VNIC  ETHER unknown 0    --    ldg1-vsw1/vnet2_ldg2
```

例 12-2 詳細なネットワークデバイス構成情報の一覧表示

次の例は、ldg1 ドメインのネットワークデバイスの詳細な一覧を `ldm list-netdev` コマンドを使用して表示します。

```
primary# ldm list-netdev -l ldg1

-----
DOMAIN
ldg1

NAME          CLASS   MEDIA   STATE   SPEED OVER LOC
-----
net0          VNET    ETHER   up       0    --    primary-vsw0/vnet0_ldg1
[/virtual-devices@100/channel-devices@200/network@0]
MTU           : 1500 [1500-1500]
IPADDR        : 10.129.241.200/255.255.255.0
MAC_ADDRS     : 00:14:4f:fb:9c:df

net3          PHYS    ETHER   up       10000 --    /SYS/MB/RISER1/PCIE4
[/pci@400/pci@1/pci@0/pci@0/network@0]
MTU           : 1500 [576-15500]
MAC_ADDRS     : a0:36:9f:0a:c5:d2

net4          VSW     ETHER   up       10000 --    ldg1-vsw1
[/virtual-devices@100/channel-devices@200/virtual-network-switch@0]
MTU           : 1500 [1500-1500]
IPADDR        : 192.168.1.2/255.255.255.0
MAC_ADDRS     : 00:14:4f:fb:61:6e

net1          PHYS    ETHER   up       10000 --    /SYS/MB/RISER1/PCIE4
[/pci@400/pci@1/pci@0/pci@0/network@0,1]
MTU           : 1500 [576-15500]
MAC_ADDRS     : a0:36:9f:0a:c5:d2

net5          VNET    ETHER   up       0    --    ldg1-vsw1/vnet1_ldg1
[/virtual-devices@100/channel-devices@200/network@1]
MTU           : 1500 [1500-1500]
IPADDR        : 0.0.0.0 /255.0.0.0
               : fe80::214:4fff:fef8:5062/ffc0::
MAC_ADDRS     : 00:14:4f:f8:50:62

net6          VNET    ETHER   up       0    --    ldg1-vsw1/vnet2_ldg1
[/virtual-devices@100/channel-devices@200/network@2]
MTU           : 1500 [1500-1500]
IPADDR        : 0.0.0.0 /255.0.0.0
               : fe80::214:4fff:fef8:af92/ffc0::
MAC_ADDRS     : 00:14:4f:f8:af:92
```

```

aggr2          AGGR    ETHER    unknown  0    net1,net3 --
MODE           : TRUNK
POLICY          : L2,L3
LACP_MODE      : ACTIVE
MEMBER         : net1 [PORTSTATE = attached]
MEMBER         : net3 [PORTSTATE = attached]
MAC_ADDRS      : a0:36:9f:0a:c5:d2

ldoms-vsw0.vport3 VNIC    ETHER    unknown  0    -- ldg1-vsw1/vnet2_ldg1
MTU            : 1500 [576-1500]
MAC_ADDRS      : 00:14:4f:f8:af:92

ldoms-vsw0.vport2 VNIC    ETHER    unknown  0    -- ldg1-vsw1/vnet1_ldg1
MTU            : 1500 [576-1500]
MAC_ADDRS      : 00:14:4f:f8:50:62

ldoms-vsw0.vport1 VNIC    ETHER    unknown  0    -- ldg1-vsw1/vnet2_ldg3
MTU            : 1500 [576-1500]
MAC_ADDRS      : 00:14:4f:f9:d3:88

ldoms-vsw0.vport0 VNIC    ETHER    unknown  0    -- ldg1-vsw1/vnet2_ldg2
MTU            : 1500 [576-1500]
MAC_ADDRS      : 00:14:4f:fa:47:f4
                : 00:14:4f:f9:65:b5
                : 00:14:4f:f9:60:3f

```

例 12-3 ネットワークデバイス統計情報の一覧表示

ldm list-netstat コマンドは、システム内の 1 つ以上のドメインのネットワーク統計情報を表示します。

次の例は、システム内のすべてのドメインのデフォルトのネットワーク統計情報を示しています。

```
primary# ldm list-netstat
```

```
DOMAIN
primary
```

NAME	IPACKETS	RBYTES	OPACKETS	OBYTES
----	-----	-----	-----	-----
net3	0	0	0	0
net0	2.72M	778.27M	76.32K	6.01M
net4	2.72M	778.27M	76.32K	6.01M
net6	2	140	1.30K	18.17K
net7	0	0	0	0
net2	0	0	0	0
net1	0	0	0	0
aggr1	0	0	0	0
ldoms-vsw0.vport0	935.40K	74.59M	13.15K	984.43K
ldoms-vsw0.vport1	933.26K	74.37M	11.42K	745.15K
ldoms-vsw0.vport2	933.24K	74.37M	11.46K	747.66K
ldoms-vsw1.vport1	202.26K	17.99M	179.75K	15.69M

ldoms-vsw1.vport0	202.37K	18.00M	189.00K	16.24M

DOMAIN				
ldg1				
NAME	IPACKETS	RBYTES	OPACKETS	OBYTES
----	-----	-----	-----	-----
net0	5.19K	421.57K	68	4.70K
net3	0	0	2.07K	256.93K
net4	0	0	4.37K	560.17K
net1	0	0	2.29K	303.24K
net5	149	31.19K	78	17.00K
net6	147	30.51K	78	17.29K
aggr2	0	0	0	0
ldoms-vsw0.vport3	162	31.69K	52	14.11K
ldoms-vsw0.vport2	163	31.74K	51	13.76K
ldoms-vsw0.vport1	176	42.99K	25	1.50K
ldoms-vsw0.vport0	158	40.19K	45	4.42K

DOMAIN				
ldg2				
NAME	IPACKETS	RBYTES	OPACKETS	OBYTES
----	-----	-----	-----	-----
net0	5.17K	418.90K	71	4.88K
net1	2.70K	201.67K	2.63K	187.01K
net2	132	36.40K	1.51K	95.07K

DOMAIN				
ldg3				
NAME	IPACKETS	RBYTES	OPACKETS	OBYTES
----	-----	-----	-----	-----
net0	5.16K	417.43K	72	4.90K
net1	2.80K	206.12K	2.67K	190.36K
net2	118	35.00K	1.46K	87.78K

仮想ネットワークデバイスで消費される物理ネットワーク帯域幅の量の制御

帯域幅リソースの制御機能を使用すると、仮想ネットワークデバイスで消費される物理ネットワーク帯域幅を制限できます。この機能は、Oracle Solaris 11 以上の OS で実行され、仮想スイッチが構成されているサービスドメインでサポートされています。Oracle Solaris 10 サービスドメインでは、ネットワーク帯域幅の設定は暗黙的に無視されます。この機能を使用すれば、1 つのゲストドメインが使用可能な物理ネットワーク帯域幅を占有することで、ほかのドメイン用に 1 つも残らないことがなくなります。

`maxbw` プロパティの値を入力することで帯域幅の制限を指定するには、`ldm add-vnet` および `ldm set-vnet` コマンドを使用します。既存の仮想ネットワークデバイスの `maxbw` プロパティ値を表示するには、`ldm list-bindings` または `ldm list-domain -o network` コマンドを使用します。最小の帯域幅の制限は 10M ビット/秒です。

ネットワーク帯域幅の制限

注記 - この機能は、ハイブリッド I/O 対応の仮想ネットワークデバイスではサポートされません。ハイブリッド I/O では、帯域幅を制限するように変更できない特定単位のハードウェアリソースが割り当てられるため、ハイブリッドモードの仮想ネットワークには `maxbw` プロパティが適用されません。仮想ネットワークデバイスの帯域幅を制限するには、ハイブリッドモードを無効にする必要があります。

帯域幅リソースの制御は、仮想スイッチを通過するトラフィックにのみ適用されます。したがって、`inter-vnet` トラフィックは、この制限の対象ではありません。物理バックエンドデバイスが構成されていない場合は、帯域幅リソースの制御を無視できます。

最低限サポートされている帯域幅の制限は、サービスマイン内の Oracle Solaris ネットワークスタックによって異なります。帯域幅の制限は、必要なだけ高い値で構成できます。上限はありません。帯域幅を制限しても、帯域幅が構成された値を超過しないことが保証されるだけです。したがって、帯域幅の制限は、仮想スイッチに割り当てられている物理ネットワークデバイスのリンク速度よりも大きい値で構成できます。

ネットワーク帯域幅の制限の設定

仮想ネットワークデバイスを作成し、`maxbw` プロパティの値を入力することで帯域幅の制限を指定するには、`ldm add-vnet` コマンドを使用します。

```
primary# ldm add-vnet maxbw=limit if-name vswitch-name domain-name
```

既存の仮想ネットワークデバイスで帯域幅の制限を指定するには、`ldm set-vnet` コマンドを使用します。

```
primary# ldm set-vnet maxbw=limit if-name domain-name
```

`maxbw` プロパティに空白の値を指定すれば、帯域幅の制限をクリアすることもできます。

```
primary# ldm set-vnet maxbw= if-name domain-name
```

次の例は、ldm コマンドを使用して帯域幅の制限を指定する方法を示しています。帯域幅は、単位付きの整数として指定されます。単位は、メガビット/秒の場合は M、ギガビット/秒の場合は G です。単位を指定しない場合は、メガビット/秒の単位になります。

例 12-4 仮想ネットワークデバイスの作成時における帯域幅の制限の設定

次のコマンドは、帯域幅の制限が 100M ビット/秒の仮想ネットワークデバイス (vnet0) を作成します。

```
primary# ldm add-vnet maxbw=100M vnet0 primary-vsw0 ldg1
```

次のコマンドでは、帯域幅の制限を最小値 (10M ビット/秒) 未満に設定しようとすると、エラーメッセージが発行されます。

```
primary# ldm add-vnet maxbw=1M vnet0 primary-vsw0 ldg1
```

例 12-5 既存の仮想ネットワークデバイスにおける帯域幅の制限の設定

次のコマンドは、既存の vnet0 デバイスでの帯域幅の制限を 200M ビット/秒に設定します。

リアルタイムのネットワークトラフィックパターンによっては、帯域幅の量が指定された 200M ビット/秒の制限に達しない可能性があります。たとえば、帯域幅が 200M ビット/秒の制限を超過しない 95M ビット/秒になる可能性があります。

```
primary# ldm set-vnet maxbw=200M vnet0 ldg1
```

次のコマンドは、既存の vnet0 デバイスでの帯域幅の制限を 2G ビット/秒に設定します。

MAC 層には帯域幅の上限がないため、ベースとなる物理ネットワークの速度が 2G ビット/秒未満である場合でも、制限を 2G ビット/秒に設定できます。このような場合、帯域幅の制限は無効になります。

```
primary# ldm set-vnet maxbw=2G vnet0 ldg1
```

例 12-6 既存の仮想ネットワークデバイスにおける帯域幅の制限のクリア

次のコマンドは、指定した仮想ネットワークデバイス (vnet0) での帯域幅の制限をクリアします。この値をクリアすると、仮想ネットワークデバイスでは、ベースとなる物理デバイスで提供されている最大限の帯域幅が使用されます。

```
primary# ldm set-vnet maxbw= vnet0 ldg1
```


例 12-7 既存の仮想ネットワークデバイスにおける帯域幅の制限の表示

`ldm list-bindings` コマンドは、指定した仮想ネットワークデバイスの `maxbw` プロパティ値 (定義されている場合) を表示します。

次のコマンドは、仮想ネットワークデバイス (`vnet0`) の帯域幅の制限が 15M ビット/秒であることを表示します。帯域幅の制限が設定されていない場合は、`MAXBW` フィールドは空白です。

```
primary# ldm list-bindings
...
VSW
NAME          MAC          NET-DEV  ID  DEVICE  LINKPROP
primary-vsw0  00:14:4f:f9:95:97 net0      0  switch@0  1

DEFAULT-VLAN-ID PVID VID      MTU  MODE  INTER-VNET-LINK
1              1      1500    on

PEER          MAC          PVID VID MTU  MAXBW  LINKPROP INTERVNETLINK
vnet0@ldg1  00:14:4f:fb:b8:c8 1      1500  15

...

NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1          bound     - - - - - 5000   8      2G

NETWORK
NAME          SERVICE          ID  DEVICE
vnet0         primary-vsw0@primary  0  network@0

MAC          MODE  PVID VID      MTU  MAXBW  LINKPROP
00:14:4f:fb:b8:c8 1      1500  15

PEER          MAC          MODE  PVID VID
primary-vsw0@primary  00:14:4f:f9:95:97 1

MTU  MAXBW  LINKPROP
1500
```

`dladm show-linkprop` コマンドを使用すると、次のように `maxbw` プロパティ値を表示することもできます。

```
# dladm show-linkprop -p maxbw
LINK          PROPERTY PERM VALUE  EFFECTIVE DEFAULT POSSIBLE
...
ldoms-vsw0.vport0 maxbw    rw   15    15      --      --
```

仮想デバイス識別子およびネットワークインタフェース名

ドメインに仮想スイッチまたは仮想ネットワークデバイスを追加する場合、`id` プロパティを設定することでデバイス番号を指定できます。

```
primary# ldm add-vsw [id=switch-id] vswitch-name domain-name
primary# ldm add-vnet [id=network-id] if-name vswitch-name domain-name
```

ドメインの各仮想スイッチおよび仮想ネットワークデバイスには、ドメインがバインドされるときに割り当てられる一意のデバイス番号があります。`id` プロパティを設定して仮想スイッチまたは仮想ネットワークデバイスを明示的なデバイス番号で追加した場合、指定したデバイス番号が使用されます。デバイス番号を指定しなかった場合、使用可能なもっとも小さいデバイス番号が自動的に割り当てられます。その場合、割り当てられるデバイス番号は、仮想スイッチまたは仮想ネットワークデバイスがシステムに追加された方法によって異なります。仮想スイッチまたは仮想ネットワークデバイスに最終的に割り当てられたデバイス番号は、ドメインがバインドされるときに `ldm list-bindings` コマンドの出力で確認できます。

次の例は、`primary` ドメインに 1 つの仮想スイッチ `primary-vsw0` が構成されていることを示しています。この仮想スイッチのデバイス番号は `0` (`switch@0`) です。

```
primary# ldm list-bindings primary
...
VSW
  NAME          MAC          NET-DEV DEVICE  DEFAULT-VLAN-ID PVID VID MTU MODE
  primary-vsw0  00:14:4f:fb:54:f2 net0    switch@0  1              1   5,6 1500
...
```

次の例は、`ldg1` ドメインには 2 つの仮想ネットワークデバイス `vnet` および `vnet1` が構成されていることを示しています。デバイス `vnet` のデバイス番号は `0` (`network@0`) で、デバイス `vnet1` のデバイス番号は `1` (`network@1`) です。

```
primary# ldm list-bindings ldg1
...
NETWORK
  NAME  SERVICE          DEVICE  MAC          MODE  PVID VID MTU
  vnet  primary-vsw0@primary network@0 00:14:4f:fb:e0:4b hybrid 1      1500
  ...
  vnet1 primary-vsw0@primary network@1 00:14:4f:f8:e1:ea      1      1500
...
```

同様に、仮想ネットワークデバイスが構成されたドメインで Oracle Solaris OS を実行している場合、仮想ネットワークデバイスはネットワークインタフェース `vnetN` を備えています。ただし、仮

想ネットワークデバイスのネットワークインタフェース番号 N は、仮想ネットワークデバイスのデバイス番号 n と同じとはかぎりません。

注記 - Oracle Solaris 11 システムでは、汎用リンクが `netn` の形式で `vsw` と `vnetn` の両方に割り当てられます。`dladm show-phys` コマンドを使用して、`vsw` デバイスと `vnetn` デバイ스에 マップされた `netn` 名を識別します。



注意 - Oracle Solaris OS では、ネットワークインタフェースの名前と、仮想スイッチまたは仮想ネットワークデバイスとの間のマッピングが、デバイス番号に基づいて保存されます。デバイス番号が仮想スイッチまたは仮想ネットワークデバイスに明示的に割り当てられていない場合、ドメインのバインドがいったん解除されたあとでふたたびバインドされると、デバイス番号が変更されることがあります。その場合、ドメインで動作している OS によって割り当てられたネットワークインタフェース名が変更され、システムの既存の構成が使用できなくなることもあります。この状況は、たとえば、仮想スイッチまたは仮想ネットワークインタフェースがドメインの構成から削除されたときに起こる場合があります。

`ldm list-*` コマンドを使用して、仮想スイッチまたは仮想ネットワークデバイスに対応する Oracle Solaris OS のネットワークインタフェース名を直接判定することはできません。ただし、`ldm list l` コマンドの出力と、Oracle Solaris OS の `/devices` 配下のエントリを組み合わせると、この情報を取得できます。

Oracle Solaris 11 ネットワークインタフェース名の検索

Oracle Solaris 11 システムでは、`ldm list-netdev` コマンドを使用して Oracle Solaris OS ネットワークインタフェース名を検索できます。詳細については、[ldm\(1M\)](#) のマニュアルページを参照してください。

次の例は、`ldm list-netdev` コマンドと `ldm list -o network` コマンドを示しています。`ldm list -o network` コマンドは、仮想ネットワークデバイスを `NAME` フィールドに示します。`ldm list-netdev` 出力は、対応する OS インタフェース名を `NAME` 列に示します。

```
primary# ldm list -o network ldg1
....
NETWORK
  NAME          SERVICE          ID DEVICE    MAC          MODE
  PVID VID MTU   MAXBW LINKPROP
  vnet0-ldg1    primary-vsw0@primary 0   network@0  00:14:4f:fa:eb:4e 1
                        1500
  vnet1-ldg1    svcdom-vsw0@svcdom  1   network@1  00:14:4f:f8:53:45 4
                        1500
                        PVLAN :400,community

primary# ldm list-netdev ldg1
```

```
DOMAIN
ldg1
```

NAME	CLASS	MEDIA	STATE	SPEED	OVER	LOC
net0	VNET	ETHER	up	0	vnet0	primary-vsw0/vnet0-ldg1
net1	VNET	ETHER	up	0	vnet1	svcdom-vsw0/vnet1-ldg1
net2	VNET	ETHER	unknown	0	vnet2	svcdom-vsw1/vnet2-ldg1

ldm list-netdev 出力が正しいことを確認するには、dladm show-phys コマンドと dladm show-linkprop -p mac-address コマンドを ldg1 から実行します。

```
ldg1# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	0	unknown	vnet0
net1	Ethernet	up	0	unknown	vnet1
net2	Ethernet	unknown	0	unknown	vnet2

```
ldg1# dladm show-linkprop -p mac-address
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net0	mac-address	rw	0:14:4f:fa:eb:4e	0:14:4f:fa:eb:4e	0:14:4f:fa:eb:4e	--
net1	mac-address	rw	0:14:4f:f8:53:45	0:14:4f:f8:53:45	0:14:4f:f8:53:45	--

▼ Oracle Solaris OS ネットワークインタフェース名を検索する方法

この手順では、net-c に対応する ldg1 内の Oracle Solaris OS ネットワークインタフェース名を検索する方法について説明します。この例では、仮想ネットワークデバイスではなく仮想スイッチのネットワークインタフェース名を検索する場合の相違点も示します。次の例では、ゲストドメイン ldg1 には net-a および net-c の 2 つの仮想ネットワークデバイスが含まれています。

1. ldm コマンドを使用して、net-c の仮想ネットワークデバイス番号を探します。

```
primary# ldm list -l ldg1
...
NETWORK
NAME          SERVICE          DEVICE          MAC
net-a         primary-vsw0@primary  network@0      00:14:4f:f8:91:4f
net-c         primary-vsw0@primary  network@2      00:14:4f:f8:dd:68
...
```

net-c の仮想ネットワークデバイス番号は 2 (network@2) です。

仮想スイッチのネットワークインタフェース名を確認するには、switch@n として表示される仮想スイッチデバイス番号 n を見つけます。

2. ldg1 で対応するネットワークインタフェースを検出するには、ldg1 にログインして、/devices 配下でこのデバイス番号に対するエントリを探します。

```
ldg1# uname -n
ldg1
ldg1# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
```

ネットワークインタフェース名は、コロンのあとのエントリの部分で、この場合は vnet1 です。

仮想スイッチのネットワークインタフェース名を判定するには、-name オプションの引数を virtual-network-switch@n* に置換します。次に、vswN という名前のネットワークインタフェースを探します。

3. 手順 1 の net-c に対する ldm list -l の出力に示されるように、vnet1 の MAC アドレスが 00:14:4f:f8:dd:68 であることを確認します。

■ Oracle Solaris 11 OS。

- a. vnet1 に指定するインタフェースの名前を確認します。

```
ldg1# dladm show-phys |grep vnet1
net2      Ethernet      up      0      unknown    vnet1
```

- b. net2 の MAC アドレスを確認します。

```
# dladm show-linkprop -p mac-address net2
LINK PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT POSSIBLE
net2 mac-address rw  00:14:4f:f8:dd:68 00:14:4f:f8:dd:68 --      --
```

この MAC アドレスの例は、手順 1 にある ldm list -l コマンドの net-c に関する出力に一致しています。

■ Oracle Solaris 10 OS。

```
ldg1# ifconfig vnet1
vnet1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 0.0.0.0 netmask 0
        ether 0:14:4f:f8:dd:68
```

自動または手動による MAC アドレスの割り当て

使用する予定の論理ドメイン、仮想スイッチ、および仮想ネットワークに割り当てられるだけの十分な数のメディアアクセス制御 (Media Access Control、MAC) アドレスが必要です。Logical Domains Manager から論理ドメイン、仮想ネットワーク、および仮想スイッチに自動

的に MAC アドレスを割り当てるか、割り当てられた MAC アドレスの自身のプールから手動で MAC アドレスを割り当てることができます。MAC アドレスを設定する `ldm` のサブコマンドは、`add-domain`、`add-vsw`、`set-vsw`、`add-vnet`、および `set-vnet` です。これらのサブコマンドで MAC アドレスを指定しない場合は、Logical Domains Manager が自動的に MAC アドレスを割り当てます。

Logical Domains Manager に MAC アドレスの割り当てを実行させる利点は、論理ドメインで使用するための専用の MAC アドレスのブロックを利用できることです。また、Logical Domains Manager は、同じサブネットにあるほかの Logical Domains Manager インスタンスと競合する MAC アドレスを検出し、これを回避します。この動作により、手動で MAC アドレスのプールを管理する必要がなくなります。

論理ドメインが作成されたり、ドメインにネットワークデバイスが構成されたりするとすぐに、MAC アドレスの割り当てが発生します。また、割り当ては、デバイスまたは論理ドメイン自体が削除されるまで保持されます。

ドメインに割り当てられる MAC アドレスの範囲

ドメインには、次の 512K の MAC アドレスのブロックが割り当てられています。

`00:14:4F:F8:00:00 - 00:14:4F:FF:FF:FF`

下位の 256K のアドレスは、Logical Domains Manager による MAC アドレスの自動割り当てに使用されるため、この範囲のアドレスを手動でリクエストすることはできません。

`00:14:4F:F8:00:00 - 00:14:4F:FB:FF:FF`

MAC アドレスを手動で割り当てる場合は、この範囲の上位半分を使用できます。

`00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF`

注記 - Oracle Solaris 11 で、VNIC の MAC アドレスの割り当ては、これらの範囲外のアドレスが使用されます。

自動割り当てのアルゴリズム

論理ドメインまたはネットワークデバイスの作成時に MAC アドレスを指定しない場合、Logical Domains Manager は MAC アドレスを自動的に確保して、その論理ドメインまたはネットワークデバイスに割り当てます。

この MAC アドレスを取得するために、Logical Domains Manager はアドレスの選択を繰り返し試みて、潜在的な競合がないか確認します。MAC アドレスは、この目的のために確保された 256K のアドレス範囲からランダムに選択されます。候補として選択される MAC アドレスが重複する可能性を少なくするために、MAC アドレスはランダムに選択されます。

選択されたアドレスは、ほかのシステムのその他の Logical Domains Manager に対して確認され、重複した MAC アドレスが実際に割り当てられることを防止します。使用されているアルゴリズムは、[263 ページの「重複した MAC アドレスの検出」](#)に記載されています。アドレスがすでに割り当てられている場合、Logical Domains Manager は、別のアドレスを選択し、競合を再度確認します。このプロセスは、まだ割り当てられていない MAC アドレスが見つかるか、30 秒の制限時間が経過するまで続きます。制限時間に達すると、デバイスの作成が失敗し、次のようなエラーメッセージが表示されます。

```
Automatic MAC allocation failed. Please set the vnet MAC address manually.
```

重複した MAC アドレスの検出

同じ MAC アドレスが別のデバイスに割り当てられないようにするために、Logical Domains Manager がデバイスに割り当てようとしているアドレスを含むマルチキャストメッセージを、制御ドメインのデフォルトのネットワークインタフェースを介して送信することで、Logical Domains Manager はほかのシステム上の Logical Domains Manager に確認します。MAC アドレスの割り当てを試行している Logical Domains Manager は、応答を 1 秒待機します。Oracle VM Server for SPARC が有効な別のシステムの異なるデバイスにその MAC アドレスがすでに割り当てられている場合は、そのシステムの Logical Domains Manager が対象となっている MAC アドレスを含む応答を送信します。リクエストを送信した Logical Domains Manager は応答を受け取ると、選択した MAC アドレスがすでに割り当てられていることに注意し、別のアドレスを選択して処理を繰り返します。

デフォルトでは、これらのマルチキャストメッセージは、同じサブネット上のほかのマネージャーにのみ送信されます。デフォルトの有効期限 (TTL) は 1 です。TTL は、サービス管理機能 (SMF) プロパティ `ldmd/hops` を使用して構成できます。

各 Logical Domains Manager は、次の処理を担当します。

- マルチキャストメッセージの待機
- ドメインに割り当てられた MAC アドレスの追跡
- 重複の検索
- 重複が発生しないようにするための応答

何らかの理由でシステム上の Logical Domains Manager が停止すると、Logical Domains Manager が停止している間に MAC アドレスの重複が発生する可能性があります。

論理ドメインまたはネットワークデバイスが作成されるときに MAC の自動割り当てが行われ、そのデバイスまたは論理ドメインが削除されるまで保持されます。

注記 - 論理ドメインまたはネットワークデバイスが作成されて論理ドメインが起動すると、重複した MAC アドレスの検出確認が行われます。

Oracle Solaris 10 を実行するドメインでのネットワークアダプタの使用

Oracle Solaris 10 論理ドメイン環境のサービスドメイン内で動作する仮想スイッチサービスは、GLDv3 準拠のネットワークアダプタと直接対話できます。GLDv3 に準拠していないネットワークアダプタは、これらのシステムで使用できますが、仮想スイッチと直接対話することはできません。GLDv3 に準拠していないネットワークアダプタを使用する方法については、[265 ページの「NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成」](#)を参照してください。

注記 - GLDv3 の準拠は Oracle Solaris 11 環境では問題になりません。

リンクアグリゲーションの使用の詳細は、[296 ページの「仮想スイッチでのリンクアグリゲーションの使用」](#)を参照してください。

▼ ネットワークアダプタが GLDv3 準拠かどうかを判定する方法

この手順は、Oracle Solaris 10 ドメインにのみ適用されます。

- ネットワークアダプタが GLDv3 準拠かどうかを判別します。

次の例では、ネットワークデバイス名として `bge0` を使用します。

```
# dladm show-link bge0
bge0          type: non-vlan   mtu: 1500      device: bge0
```

`type`: フィールドの値は次のいずれかになります。

- GLDv3 に準拠しているドライバの種類は、`non-vlan` または `vlan` です。
- GLDv3 に準拠していないドライバの種類は、`legacy` です。

NAT およびルーティング用の仮想スイッチおよびサービスドメインの構成

Oracle Solaris 10 OS で、仮想スイッチ (`vsw`) はレイヤー 2 スイッチで、サービスドメインでネットワークデバイスとしても使用できます。仮想スイッチは、さまざまな論理ドメインで仮想ネットワークデバイス間のスイッチとしてのみ動作するように構成できますが、物理デバイスを介してネットワークの外部に接続することはできません。このモードで、`vsw` をネットワークデバイスとして作成し、サービスドメインで IP ルーティングを有効にすると、仮想ネットワークでサービスドメインをルーターとして使用して外部と通信できます。このモードでの操作は、物理ネットワークアダプタが GLDv3 に準拠していない場合、ドメインが外部に接続できるようにするために非常に重要です。

この構成の利点は次のとおりです。

- 仮想スイッチは物理デバイスを直接使用する必要がなく、基本となるデバイスが GLDv3 に準拠していない場合でも外部と接続できます。
- この構成では、Oracle Solaris OS の IP ルーティングとフィルタリング機能を利用できます。

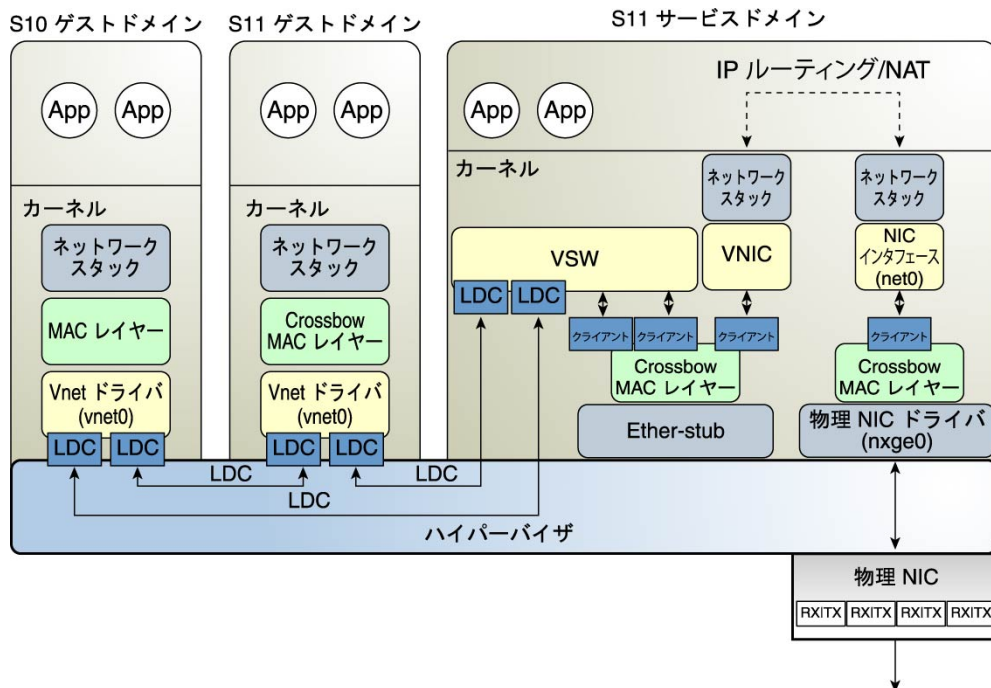
Oracle Solaris 11 システムでの NAT の構成

Oracle Solaris 11 ネットワーク仮想化機能には、擬似ネットワークデバイスである `etherstub` が含まれています。このデバイスは、物理ネットワークデバイスと同様の機能を備えていますが、そのクライアントとのプライベート通信専用です。この擬似デバイスは、仮想スイッチのネッ

トワークバックエンドデバイスとして使用でき、仮想ネットワーク間のプライベート通信を提供します。etherstub デバイスをバックエンドデバイスとして使用することで、ゲストドメインは同じ etherstub デバイス上の VNIC と通信できます。このように etherstub デバイスを使用すると、ゲストドメインはサービスドメイン内のゾーンと通信できます。dladm create-etherstub コマンドを使用して、etherstub デバイスを作成します。

次の図に、仮想スイッチ、etherstub デバイス、VNIC を使用して、サービスドメインにネットワークアドレス変換 (NAT) を設定する方法を示します。

図 12-5 仮想ネットワークルーティング



永続的なルートの使用を検討する場合があります。詳細は、『[Troubleshooting Network Administration Issues in Oracle Solaris 11.3](#)』の「[Troubleshooting Issues When Adding a Persistent Route](#)」および『[Configuring and Managing Network Components in Oracle Solaris 11.3](#)』の「[Creating Persistent \(Static\) Routes](#)」を参照してください。

▼ ドメインが外部に接続できるように仮想スイッチを設定する方法 (Oracle Solaris 11)

1. Oracle Solaris 11 etherstub デバイスを作成します。

```
primary# dladm create-etherstub stub0
```

2. `stub0` を物理バックエンドデバイスとして使用する仮想スイッチを作成します。

```
primary# ldm add-vsw net-dev=stub0 primary-stub-vsw0 primary
```

3. `stub0` デバイスに VNIC を作成します。

```
primary# dladm create-vnic -l stub0 vnic0
```

4. `vnic0` をネットワークインタフェースとして構成します。

```
primary# ipadm create-ip vnic0
primary# ipadm create-addr -T static -a 192.168.100.1/24 vnic0/v4static
```

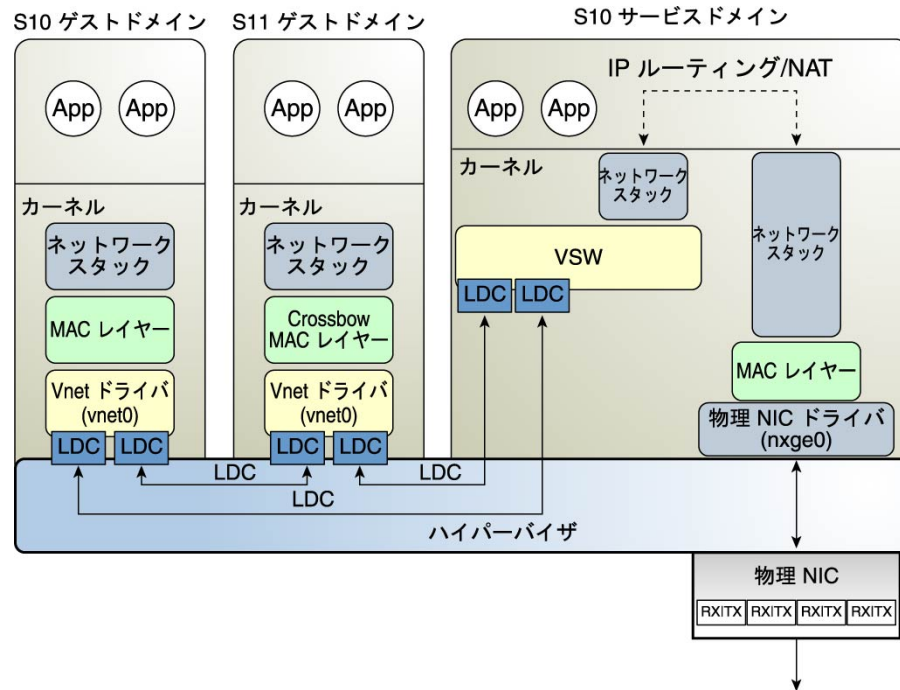
5. IPv4 転送を有効にし、NAT 規則を作成します。

[『Configuring and Managing Network Components in Oracle Solaris 11.3』](#)の「Customizing IP Interface Properties and Addresses」および[『Oracle Solaris Administration: IP Services』](#)の「Packet Forwarding and Routing on IPv4 Networks」を参照してください。

Oracle Solaris 10 システムでの NAT の構成

次の図は、仮想スイッチを使用して、サービスドメインにネットワークアドレス変換 (NAT) を構成し、ゲストドメインが外部接続できるようにする方法を示しています。

図 12-6 仮想ネットワークルーティング



▼ Oracle Solaris 10 サービスドメイン上の仮想スイッチをドメインへの外部接続を提供するように設定する方法

1. 物理デバイスが関連付けられていない仮想スイッチを作成します。
アドレスを割り当てる場合は、仮想スイッチに一意の MAC アドレスが割り当てられるようにしてください。

```
primary# ldmd add-vsw [mac-addr=xx:xx:xx:xx:xx:xx] ldg1-vsw0 ldg1
```
2. ドメインによって使用される物理ネットワークデバイスに加えて、仮想スイッチをネットワークデバイスとして作成します。
仮想スイッチの作成の詳細については、[47 ページの「仮想スイッチをプライマリインタフェースとして構成する方法」](#)を参照してください。
3. 必要に応じて、DHCP で仮想スイッチデバイスを構成します。

DHCP での仮想スイッチデバイスの構成の詳細については、[47 ページの「仮想スイッチをブライマリインタフェースとして構成する方法」](#)を参照してください。

4. 必要に応じて、`/etc/dhcp.vsw` ファイルを作成します。
5. サービスドメインで IP ルーティングを構成し、すべてのドメインで必要なルーティングテーブルを設定します。

IP ルーティングの詳細については、『[Oracle Solaris Administration: IP Services](#)』の「[Packet Forwarding and Routing on IPv4 Networks](#)」を参照してください。

Oracle VM Server for SPARC 環境での IPMP の構成

Oracle VM Server for SPARC ソフトウェアは、仮想ネットワークデバイスによるリンクベースの IP ネットワークマルチパス (IPMP) をサポートしています。仮想ネットワークデバイスで IPMP グループを構成する場合は、リンクベースの検出を使用するようにグループを構成します。Oracle VM Server for SPARC (Logical Domains) ソフトウェアの以前のバージョンを使用している場合、仮想ネットワークデバイスでプローブベースの検出のみを構成できます。

Oracle Solaris 11 ドメイン内の IPMP グループへの仮想ネットワークデバイスの構成

[図12-7「個別の仮想スイッチインスタンスに接続された 2 つの仮想ネットワーク \(Oracle Solaris 11\)」](#)は、サービスドメイン内の個別の仮想スイッチインスタンス (`vsw0` と `vsw1`) に接続された 2 つの仮想ネットワーク (`vnet0` と `vnet1`) を示しています。これらの仮想スイッチインスタンスがさらに、2 つの異なる物理インタフェースを使用しています。これらの物理インタフェースは、Oracle Solaris 11 サービスドメイン内の `net0` と `net1` です。

サービスドメインの物理リンクに障害が発生した場合、その物理デバイスにバインドされた仮想スイッチデバイスがリンクの障害を検出します。次に、仮想スイッチデバイスは、その仮想スイッチにバインドされた対応する仮想ネットワークデバイスに障害を伝播します。仮想ネットワークデバイスは、このリンクイベントの通知をゲスト LDom_A の IP 層に送信し、その結果、IPMP グループのもう一方の仮想ネットワークデバイスにフェイルオーバーします。

図 12-7 個別の仮想スイッチインスタンスに接続された 2 つの仮想ネットワーク (Oracle Solaris 11)

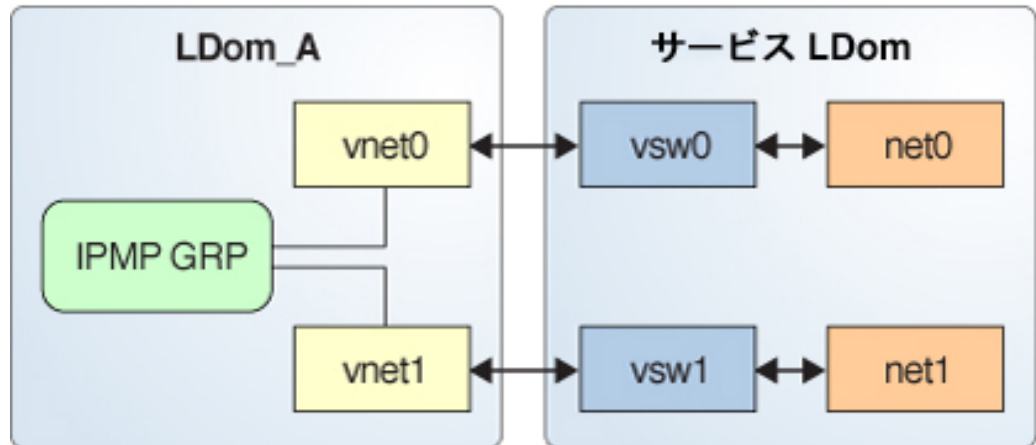
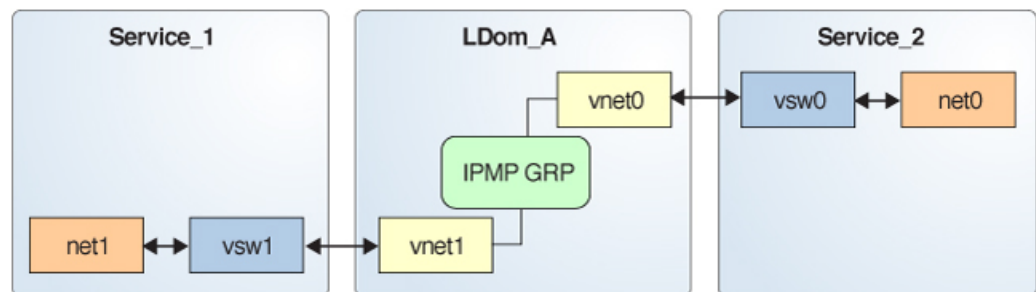


図12-8「それぞれ異なるサービスドメインに接続された複数の仮想ネットワークデバイス (Oracle Solaris 11)」は、各仮想ネットワークデバイス (vnet0 と vnet1) を異なるサービスドメイン内の仮想スイッチインスタンスに接続することによって、論理ドメイン内のさらに高い信頼性を実現できることを示しています。この場合、物理ネットワークの障害に加えて、LDom_A が仮想ネットワークの障害を検出し、サービスドメインがクラッシュまたは停止したあとでフェイルオーバーを引き起こすことができます。

図 12-8 それぞれ異なるサービスドメインに接続された複数の仮想ネットワークデバイス (Oracle Solaris 11)



詳細は、[Oracle Solaris 11.3 Information Library](#) にある「Oracle Solaris ネットワークの確立」を参照してください。

Oracle Solaris 10 ドメイン内の IPMP グループへの仮想ネットワークデバイスの構成

図12-9「個別の仮想スイッチインスタンスに接続された 2 つの仮想ネットワーク (Oracle Solaris 10)」は、サービスドメイン内の個別の仮想スイッチインスタンス (vsw0 と vsw1) に接続された 2 つの仮想ネットワーク (vnet0 と vnet1) を示しています。これらの仮想スイッチインスタンスがさらに、2 つの異なる物理インタフェースを使用しています。これらの物理インタフェースは、Oracle Solaris 10 サービスドメイン内の nxge0 と nxge1 です。

サービスドメインの物理リンクに障害が発生した場合、その物理デバイスにバインドされた仮想スイッチデバイスがリンクの障害を検出します。次に、仮想スイッチデバイスは、その仮想スイッチにバインドされた対応する仮想ネットワークデバイスに障害を伝播します。仮想ネットワークデバイスは、このリンクイベントの通知をゲスト LDom_A の IP 層に送信し、その結果、IPMP グループのもう一方の仮想ネットワークデバイスにフェイルオーバーします。

図 12-9 個別の仮想スイッチインスタンスに接続された 2 つの仮想ネットワーク (Oracle Solaris 10)

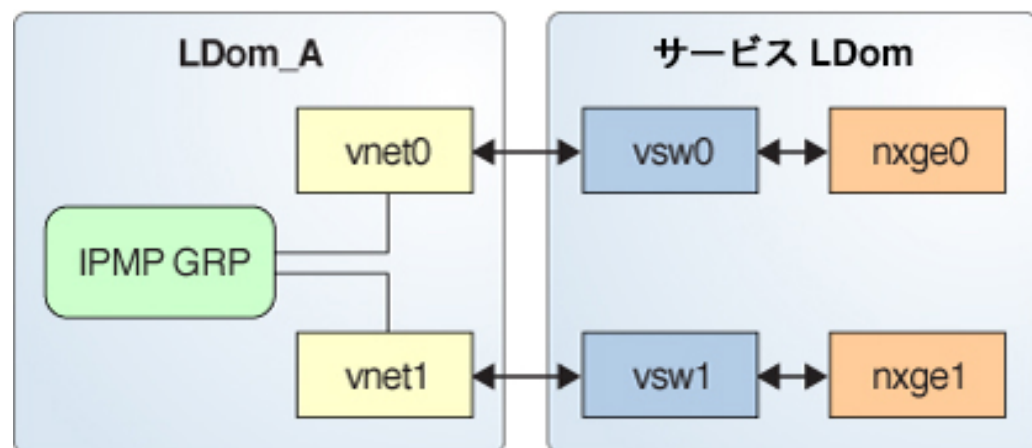
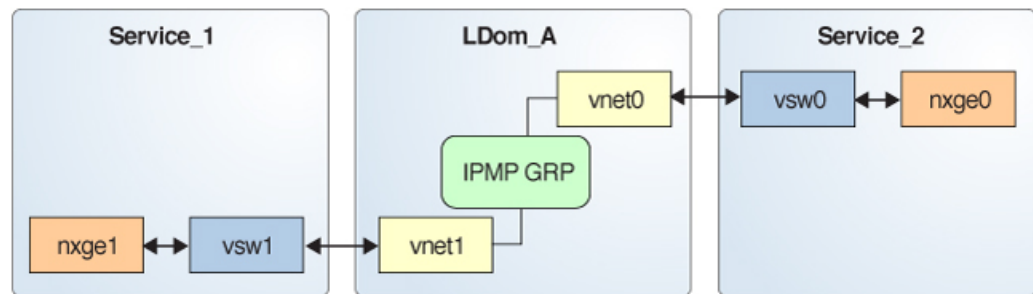


図12-10「それぞれ異なるサービスドメインに接続された複数の仮想ネットワークデバイス (Oracle Solaris 10)」は、各仮想ネットワークデバイス (vnet0 と vnet1) を異なるサービスドメイン内の仮想スイッチインスタンスに接続することによって、論理ドメイン内のさらに高い信頼性を実現できることを示しています。この場合、物理ネットワークの障害に加えて、LDom_A が仮想ネットワークの障害を検出し、サービスドメインがクラッシュまたは停止したあとでフェイルオーバーを引き起こすことができます。

図 12-10 それぞれ異なるサービスドメインに接続された複数の仮想ネットワークデバイス (Oracle Solaris 10)



詳細は、『[Oracle Solaris Administration: IP Services](#)』を参照してください。

サービスドメインでの IPMP の構成と使用

Oracle Solaris 11 システムでは、仮想ネットワークやドメインが存在しないシステムの場合と同じように物理インタフェースをグループに構成することによって、サービスドメイン内の IPMP を構成できます。Oracle Solaris 10 システムでは、仮想スイッチインタフェースをグループに構成することによって、サービスドメイン内の IPMP を構成できます。図12-11「IPMP グループの一部として構成された 2 つの物理 NIC (Oracle Solaris 11)」と図12-12「IPMP グループの一部として構成された 2 つの仮想スイッチインタフェース (Oracle Solaris 10)」は、2 つの異なる物理デバイスにバインドされた 2 つの仮想スイッチインスタンス (vsw0 と vsw1) を示しています。この場合、この 2 つの仮想スイッチインタフェースを作成して IPMP グループに構成できます。物理リンクに障害が発生した場合、その物理デバイスにバインドされた仮想スイッチデバイスがリンクの障害を検出します。次に、仮想スイッチデバイスは、このリンクイベントの通知をサービスドメインの IP 層に送信し、その結果、IPMP グループのもう一

方の仮想スイッチデバイスにフェイルオーバーします。これらの 2 つの物理インタフェースは、Oracle Solaris 11 では `net0` と `net1`、また Oracle Solaris 10 では `nxge0` と `nxge1` です。

図 12-11 IPMP グループの一部として構成された 2 つの物理 NIC (Oracle Solaris 11)

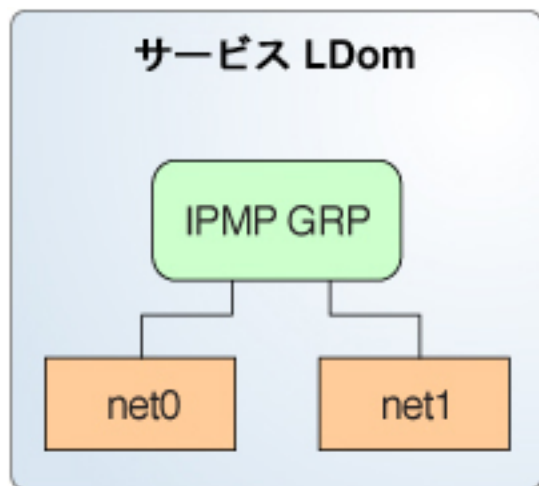
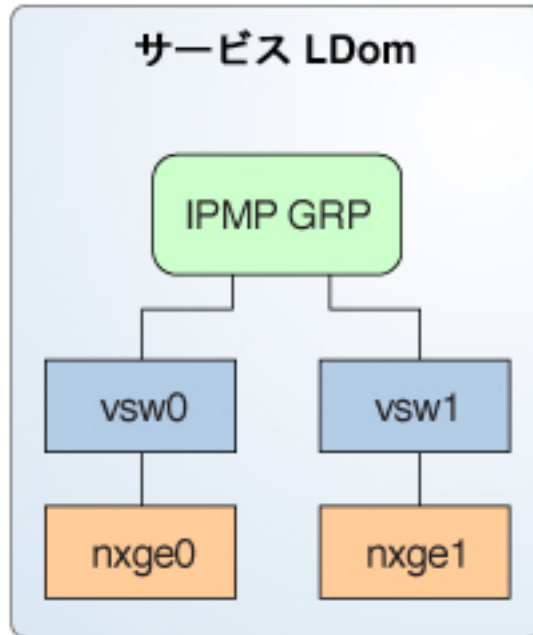


図 12-12 IPMP グループの一部として構成された 2 つの仮想スイッチインタフェース (Oracle Solaris 10)



Oracle VM Server for SPARC 仮想ネットワークでのリンクベースの IPMP の使用

仮想ネットワークおよび仮想スイッチデバイスは、リンクステータスのネットワークスタックへの更新をサポートします。デフォルトでは、仮想ネットワークデバイスはその仮想リンク (仮想スイッチへの LDC) のステータスをレポートします。この構成はデフォルトで有効になり、追加の構成手順を実行する必要はありません。

場合によっては、物理ネットワークのリンク状態の変更を検出する必要があります。たとえば、物理デバイスが仮想スイッチに割り当てられている場合、仮想ネットワークデバイスからその仮想スイッチデバイスへのリンクが動作していても、サービスドメインから外部ネットワークへの物理ネットワークリンクは停止している可能性があります。このような場合、物理リンクステータスを取得して仮想ネットワークデバイスとそのスタックにレポートする必要がある可能性があります。

`linkprop=phys-state` オプションを使用すると、仮想ネットワークデバイスおよび仮想スイッチデバイスに対して物理リンク状態の追跡を構成できます。このオプションを有効にすると、仮想デバイス（仮想ネットワークまたは仮想スイッチ）が、ドメインでインタフェースとして作成されている間、物理リンクステータスに基づいてリンクステータスをレポートします。`dladm`、`ifconfig` などの、Oracle Solaris の標準ネットワーク管理コマンドを使用して、リンクステータスを確認できます。また、リンクステータスは `/var/adm/messages` ファイルにも記録されます。

Oracle Solaris 10 の場合、[dladm\(1M\)](#) および [ifconfig\(1M\)](#) のマニュアルページを参照してください。Oracle Solaris 11 の場合、[dladm\(1M\)](#)、[ipadm\(1M\)](#)、[ipmpstat\(1M\)](#) のマニュアルページを参照してください。

注記 - 1 つの Oracle VM Server for SPARC システムで、リンクステータスを認識しないものとリンクステータスを認識するものの両方の `vnet` および `vsw` ドライバを同時に実行できます。ただし、リンクベースの IPMP を構成する場合、リンクステータスを認識するドライバをインストールする必要があります。物理リンク状態の更新を有効にする場合、`vnet` および `vsw` の両方のドライバを Oracle Solaris 10 1/13 OS にアップグレードして、Logical Domains Manager の version 1.3 以上を実行します。

▼ 物理リンクのステータスの更新を構成する方法

この手順では、仮想ネットワークデバイスで物理リンクステータスの更新を有効にする方法を示します。

同様の手順に従い、`ldm add-vsw` および `ldm set-vsw` コマンドに `linkprop=phys-state` オプションを指定することで、仮想スイッチデバイスで物理リンクステータスの更新を有効にすることもできます。

注記 - `linkprop=phys-state` オプションは、仮想スイッチデバイス自体がインタフェースとして作成されている場合にのみ使用する必要があります。`linkprop=phys-state` が指定され、物理リンクが停止している場合、仮想スイッチへの接続が有効であっても、仮想ネットワークデバイスはリンクステータスを停止状態とレポートします。この状況が発生するのは、Oracle Solaris OS は現在、仮想リンクステータスと物理リンクステータスなど、2 つの異なるリンクステータスをレポートするインタフェースを備えていないためです。

1. 管理者になります。

Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「[About Using Rights to Control Users and Processes](#)」を参照してください。

2. 仮想デバイスで物理リンクステータスの更新を有効にします。

仮想ネットワークデバイスで物理リンクステータスの更新を有効にするには、次の手順に従います。

- `ldm add-vnet` コマンド実行時に `linkprop=phys-state` を指定し、仮想ネットワークデバイスを作成します。

`linkprop=phys-state` オプションを指定すると、仮想ネットワークデバイスが物理リンクステータスの更新を取得してスタックにレポートするように構成されます。

注記 - `linkprop=phys-state` が指定され、物理リンクが停止している場合、仮想スイッチへの接続が有効であっても、仮想ネットワークデバイスはリンクステータスを `down` とレポートします。この状況が発生するのは、Oracle Solaris OS は現在、仮想リンクステータスと物理リンクステータスなど、2 つの異なるリンクステータスをレポートするインタフェースを備えていないためです。

```
primary# ldm add-vnet linkprop=phys-state if-name vswitch-name domain-name
```

次の例では、論理ドメイン `ldom1` の `primary-vsw0` に接続された `ldom1_vnet0` で物理リンクステータスの更新を有効にします。

```
primary# ldm add-vnet linkprop=phys-state ldom1_vnet0 primary-vsw0 ldom1
```

- `ldm set-vnet` コマンド実行時に `linkprop=phys-state` を指定し、既存の仮想ネットワークデバイスを変更します。

```
primary# ldm set-vnet linkprop=phys-state if-name domain-name
```

次の例では、論理ドメイン `ldom1` の `vnet0` で物理リンクステータスの更新を有効にします。

```
primary# ldm set-vnet linkprop=phys-state ldom1_vnet0 ldom1
```

物理リンクステータスの更新を無効にするには、`ldm set-vnet` コマンドを実行して `linkprop=` を指定します。

次の例では、論理ドメイン `ldom1` の `ldom1_vnet0` で物理リンクステータスの更新を無効にします。

```
primary# ldm set-vnet linkprop= ldom1_vnet0 ldom1
```

例 12-8 リンクベースの IPMP の構成

次の例は、物理リンクステータスの更新を有効にする方法と有効にしない方法の両方を使用してリンクベースの IPMP を構成する方法を示します。

- 次の例では、1 つのドメインで 2 つの仮想ネットワークデバイスを構成します。各仮想ネットワークデバイスは、リンクベースの IPMP を使用するためにサービスドメインの個別の仮想スイッチデバイスに接続されます。

注記 - これらの仮想ネットワークデバイスでテストアドレスは構成されません。また、`ldm add-vnet` コマンドを使用してこれらの仮想ネットワークデバイスを作成する場合に、追加構成を実行する必要はありません。

次のコマンドは、仮想ネットワークデバイスをドメインに追加します。`linkprop=phys-state` が指定されていないため、仮想スイッチへのリンクのみでステータスの変更がモニターされることに注意してください。

```
primary# ldm add-vnet ldom1_vnet0 primary-vsw0 ldom1
primary# ldm add-vnet ldom1_vnet1 primary-vsw1 ldom1
```

次のコマンドは、仮想ネットワークデバイスをゲストドメインで構成して IPMP グループに割り当てます。リンクベースの障害検出が使用されているためにこれらの仮想ネットワークデバイスでテストアドレスが構成されていないことに注意してください。

- **Oracle Solaris 10 OS:** `ifconfig` コマンドを使用します。

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

2 番目および 3 番目のコマンドは、必要に応じて、IP アドレスとの `ipmp0` インタフェースを構成します。

- **Oracle Solaris 11 OS:** `ipadm` コマンドを使用します。

`net0` と `net1` はそれぞれ `vnet0` と `vnet1` の Oracle Solaris 11 バニティー名です。

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 ipmp0
```

- 次の例では、1 つのドメインで 2 つの仮想ネットワークデバイスを構成します。各ドメインは、リンクベースの IPMP を使用するためにサービスドメインの個別の仮想スイッチデバイスに接続されます。また、仮想ネットワークデバイスは、物理リンクステータスの更新を取得するように構成されます。

```
primary# ldm add-vnet linkprop=phys-state ldom1_vnet0 primary-vsw0 ldom1
primary# ldm add-vnet linkprop=phys-state ldom1_vnet1 primary-vsw1 ldom1
```

注記 - ドメインを正常にバインドするために、仮想スイッチに物理ネットワークデバイスを割り当てる必要があります。ドメインがすでにバインドされており、仮想スイッチに物理ネットワークデバイスが割り当てられていない場合、ldm add-vnet コマンドは失敗します。

次のコマンドは、仮想ネットワークデバイスを作成して IPMP グループに割り当てます。

- **Oracle Solaris 10 OS:** ifconfig コマンドを使用します。

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 192.168.1.1/24 up
# ifconfig vnet1 192.168.1.2/24 up
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

- **Oracle Solaris 11 OS:** ipadm コマンドを使用します。

net0 と net1 はそれぞれ vnet0 と vnet1 のバニティー名です。

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 ipmp0
# ipadm create-addr -T static -a 192.168.1.1/24 ipmp0/v4addr1
# ipadm create-addr -T static -a 192.168.1.2/24 ipmp0/v4addr2
```

VLAN のタグ付けの使用

Oracle VM Server for SPARC ソフトウェアは、ネットワークインフラストラクチャーの 802.1Q VLAN タグ付けをサポートしています。

仮想スイッチ (vsw) および仮想ネットワーク (vnet) デバイスは、仮想ローカルエリアネットワーク (VLAN) 識別子 (ID) に基づいて Ethernet パケットのスイッチングをサポートし、Ethernet フレームの必要なタグ付けまたはタグなし処理を行います。

ゲストドメインの仮想ネットワークデバイスに複数の VLAN インタフェースを作成できます。仮想ネットワークデバイスに VLAN インタフェースを作成するには、Oracle Solaris 10 の `ifconfig` コマンドまたは Oracle Solaris 11 の `dladm` コマンドと `ipadm` コマンドを使用します。作成方法は、他の任意の物理ネットワークデバイスに VLAN インタフェースを構成する場合に使用する方法と同じです。Oracle VM Server for SPARC 環境での追加の要件として、`ldm` コマンドを使用して、VLAN を `vsw` または `vnet` 仮想ネットワークデバイスに割り当てる必要があります。[ldm\(1M\)](#) マニュアルページを参照してください。

同様に、Oracle Solaris 10 サービスドメイン内の仮想スイッチデバイス上に VLAN インタフェースを構成できます。VLAN ID 2 - 4094 が有効です。VLAN ID 1 は、`default-vlan-id` として予約されています。Oracle Solaris 11 サービスドメイン内の仮想スイッチ上に VLAN ID を構成する必要はありません。

ゲストドメインに仮想ネットワークデバイスを作成する場合、`ldm add-vnet` コマンドに `pvid=` および `vid=` 引数を使用して、ポート VLAN ID および 0 個以上のこの仮想ネットワークの VLAN ID を指定することにより、必要な VLAN にその仮想ネットワークデバイスを割り当てる必要があります。この情報によって、仮想スイッチは、Oracle VM Server for SPARC ネットワークで複数の VLAN をサポートし、ネットワークで MAC アドレスと VLAN ID の両方を使用してパケットをスイッチングするように構成されます。

Oracle Solaris 10 サービスドメインによって使用される VLAN はすべて、`vsw` デバイス上に構成する必要があります。`pvid` および `vid` プロパティ値を指定するには、`ldm add-vsw` または `ldm set-vsw` コマンドを使用します。

デバイスが属する VLAN は、`ldm set-vnet` または `ldm set-vsw` コマンドを使用して変更できます。

ポート VLAN ID

ポート VLAN ID (PVID) は、仮想ネットワークデバイスをメンバーにする必要のある VLAN を、タグなしモードで指定します。この場合、PVID で指定した VLAN の `vnet` デバイスのために必要なフレームのタグ付けまたはタグなし処理は、`vsw` デバイスによって行われます。仮想ネットワークからのタグなしのアウトバウンドフレームは、仮想スイッチによって PVID でタグ付けされます。この PVID でタグ付けされたインバウンドフレームは、仮想スイッチによってタグが削除されてから、`vnet` デバイスに送信されます。このため、PVID を仮想ネットワークに暗黙に割り当てることは、仮想スイッチの対応する仮想ネットワークポートが、PVID で指定された VLAN に対してタグなしとしてマークされることを意味します。仮想ネットワークデバイスに設定できる PVID は 1 つだけです。

対応する仮想ネットワークインタフェースを、VLAN ID を使用せずにそのデバイスインスタンスだけで構成した場合、このインタフェースは仮想ネットワークの PVID によって指定された VLAN に暗黙に割り当てられます。

たとえば、次のコマンドを使用して仮想ネットワークインスタンス `0` を作成する場合、および `vnet` の `pvid=` 引数が `10` として指定されている場合、`vnet0` インタフェースが VLAN 10 に属するように暗黙的に割り当てられます。次のコマンドは、Oracle Solaris 10 に属する `vnet0` インタフェース名を表示します。Oracle Solaris 11 の場合、代わりに `net0` などの汎用名を使用します。

- **Oracle Solaris 10 OS:** `ifconfig` コマンドを使用します。

```
# ifconfig vnet0 plumb
```

- **Oracle Solaris 11 OS:** `ipadm` コマンドを使用します。

```
# ipadm create-ip net0
```

VLAN ID

VID ID (VID) は、仮想ネットワークデバイスまたは仮想スイッチをメンバーにする必要のある VLAN を、タグ付きモードで示します。仮想ネットワークデバイスは、その VID で指定されている VLAN でタグ付きフレームを送受信します。仮想スイッチは、仮想ネットワークデバイスと外部ネットワークの間で、指定の VID でタグ付けされたフレームを通過させます。

VLAN の割り当ておよび使用

次のタスクで使用されるデバイスの例では、ドメイン内でインスタンス番号 0 を使用します。各 VLAN は、次のサブネットにマップされます。

- VLAN 20 サブネット 192.168.1.0 (ネットマスク: 255.255.255.0)
- VLAN 21 サブネット 192.168.2.0 (ネットマスク: 255.255.255.0)
- VLAN 22 サブネット 192.168.3.0 (ネットマスク: 255.255.255.0)

▼ Oracle Solaris 11 サービスドメイン内で VLAN を割り当てて使用する方法

1. 仮想スイッチ (vsw) を割り当てます。

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

2. サービスドメイン内に VLAN インタフェースを作成します。

ipadm create-addr コマンドの -T static オプションは、Oracle Solaris 11.1 OS より古い Oracle Solaris 11 OS を実行している場合にのみ必要であることに注意してください。Oracle Solaris 11 OS 以降、-T static はデフォルトの動作です。

```
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.168.2.100/24 net0
# dladm create-vlan -l net0 -v 20 vlan20
# ipadm create-ip vlan20
# ipadm create-addr -T static -a 192.168.2.100/24 vlan20
```

Oracle Solaris 11 OS で VLAN インタフェースを構成する方法の詳細については、『[Managing Network Datalinks in Oracle Solaris 11.3](#)』の第 3 章「[Configuring Virtual Networks by Using Virtual Local Area Networks](#)」を参照してください。

▼ Oracle Solaris 10 サービスドメイン内で VLAN を割り当てて使用する方法

1. 仮想スイッチ (vsw) を 2 つの VLAN に割り当てます。

たとえば、VLAN 21 をタグなし、VLAN 20 をタグ付きとして構成します。サービスドメインは VLAN ID 22 にアクセスするように構成されないことに注意してください。

```
primary# ldm add-vsw net-dev=nxge0 pvid=21 vid=20 primary-vsw0 primary
```

2. サービスドメイン内に VLAN インタフェースを作成します。

```
# ifconfig vsw0 plumb
# ifconfig vsw0 192.168.2.100 netmask 0xffffffff broadcast + up
# ifconfig vsw20000 plumb
# ifconfig vsw20000 192.168.1.100 netmask 0xffffffff broadcast + up
```

▼ Oracle Solaris 11 ゲストドメイン内で VLAN を割り当てて使用する方法

このタスクを完了すると、ldom1 ゲストドメインは、primary サービスドメインや、外部でタグ付けされた VLAN ID 21 および 192.168.2.0/24 上の IP アドレスを使用するリモートシステムや外部システムと通信できます。ldom1 ゲストドメインはまた、タグ付き VLAN ID 20 および 192.168.1.0/24 上の IP アドレスを使用するサービスドメインおよび外部システムとも通信できます。ldom1 ゲストドメインは、VLAN 22 および 192.168.3.0/24 上の IP アドレスを使用する外部システムとのみ通信でき、それらを使用するサービスドメインとは通信できません。

1. 仮想ネットワーク (vnet) を 2 つの VLAN に割り当てます。

たとえば、VLAN 21 をタグなし、VLAN 20 をタグ付きとして構成します。

```
primary# ldm add-vnet pvid=21 vid=20,22 vnet0 primary-vsw0 ldom1
ldom1# ipadm create-ip net0
ldom1# ipadm create-addr -t 192.168.2.101/24 net0
```

2. ゲストドメイン内に VLAN インタフェースを作成します。

```
ldom1# dladm create-vlan -l net0 -v 20 vlan20
ldom1# ipadm create-ip vlan20
ldom1# ipadm create-addr -t 192.168.1.101/24 vlan20

ldom1# dladm create-vlan -l net0 -v 22 vlan22
ldom1# ipadm create-ip vlan22
ldom1# ipadm create-addr -t 192.168.3.101/24 vlan22
```

▼ Oracle Solaris 10 ゲストドメイン内で VLAN を割り当てて使用する方法

このタスクを完了すると、ldom1 ゲストドメインは、primary サービスドメインや、外部でタグ付けされた VLAN ID 21 および 192.168.2.0/24 上の IP アドレスを使用するリモートシステムや外部システムと通信できます。ldom1 ゲストドメインはまた、タグ付き VLAN ID 20 および 192.168.1.0/24 上の IP アドレスを使用するサービスドメインおよび外部システムとも通信で

きます。ldom1 ゲストドメインは、VLAN 22 および 192.168.3.0/24 上の IP アドレスを使用する外部システムとのみ通信でき、それらを使用するサービスドメインとは通信できません。

1. 仮想ネットワーク (vnet) を 2 つの VLAN に割り当てます。

たとえば、VLAN 21 をタグなし、VLAN 20 をタグ付きとして構成します。

```
primary# ldm add-vnet pvid=21 vid=20,22 vnet0 primary-vsw0 ldom1
ldom1# ifconfig vnet0 plumb
ldom1# ifconfig vnet0 192.168.2.101 netmask 0xffffffff00 broadcast + up
```

2. ゲストドメイン内に VLAN インタフェースを作成します。

```
ldom1# ifconfig vnet20000 plumb
ldom1# ifconfig vnet20000 192.168.1.102 netmask 0xffffffff00 broadcast + up
ldom1# ifconfig vnet22000 plumb
ldom1# ifconfig vnet22000 192.168.3.102 netmask 0xffffffff00 broadcast + up
```

▼ インストールサーバーが VLAN に存在する場合にゲストドメインをインストールする方法

インストールサーバーが VLAN 内に存在するとき、Oracle Solaris JumpStart 機能を使用してネットワーク経由でゲストドメインをインストールする場合は注意してください。この機能は Oracle Solaris 10 システムでのみサポートされています。

Oracle Solaris JumpStart 機能を使用してゲストドメインをインストールする方法については、[59 ページの「Oracle Solaris 10 ゲストドメインで Oracle Solaris JumpStart 機能を使用する方法」](#)を参照してください。

1. ネットワークデバイスをタグなしモードで構成します。

たとえば、インストールサーバーが VLAN 21 にある場合、最初に仮想ネットワークを次のように構成します。

```
primary# ldm add-vnet pvid=21 vnet01 primary-vsw0 ldom1
```

その仮想ネットワークデバイスにタグ付き VLAN (vid) を構成しないでください。OpenBoot PROM (OBP) は VLAN を認識せず、VLAN のタグ付きのネットワークパケットを処理できないため、このようにする必要があります。

2. インストールが完了して Oracle Solaris OS がブートしたら、仮想ネットワークをタグ付きモードで構成します。

```
primary# ldm set-vnet pvid= vid=21, 22, 23 vnet01 primary-vsw0 ldom1
```

これで、仮想ネットワークデバイスをタグ付きモードでほかの VLAN に追加できます。

プライベート VLAN の使用

プライベート VLAN (PVLAN) のメカニズムを使用すると、通常の VLAN をサブ VLAN に分割することによってネットワークトラフィックを隔離できます。PVLAN のメカニズムは、[RFC 5517 \(http://tools.ietf.org/html/rfc5517\)](http://tools.ietf.org/html/rfc5517) で定義されています。通常の VLAN は通常、1 つのブロードキャストドメインですが、PVLAN のプロパティを使用して構成されると、既存のレイヤー 3 構成を維持したまま、1 つのブロードキャストドメインがより小さいブロードキャストサブドメインにパーティション化されます。PVLAN を構成する場合、通常の VLAN はプライマリ VLAN と呼ばれ、サブ VLAN はセカンダリ VLAN と呼ばれます。

2 つの仮想ネットワークが物理リンク上の同じ VLAN ID を使用する場合は、すべてのブロードキャストトラフィックがこの 2 つの仮想ネットワークの間で渡されます。ただし、PVLAN のプロパティを使用する仮想ネットワークを作成した場合は、パケット転送動作がすべての状況には適用されない可能性があります。

次の表に、隔離 PVLAN とコミュニティー PVLAN でのブロードキャストパケット転送のルールを示します。

表 12-1 ブロードキャストパケット転送のルール

PVLAN タイプ	隔離	コミュニティー A	コミュニティー B
隔離	いいえ	いいえ	いいえ
コミュニティー A	いいえ	はい	いいえ
コミュニティー B	いいえ	いいえ	はい

たとえば、vnet0 仮想ネットワークと vnet1 仮想ネットワークの両方が net0 ネットワーク上で分離されている場合、net0 は 2 つの仮想ネットワーク間でブロードキャストトラフィックを渡しません。ただし、net0 ネットワークが分離された VLAN からトラフィックを受信する場合、トラフィックは VLAN に関連する分離されたポートには渡されません。この状況は、分離された仮想ネットワークがプライマリ VLAN からのトラフィックのみを受け付けるために発生します。

inter-vnet リンク機能は、隔離 PVLAN とコミュニティー PVLAN の通信制限をサポートします。inter-vnet リンクは、隔離 PVLAN では無効であり、コミュニティー PVLAN の場合は同

じコミュニティ内に存在する仮想ネットワークでのみ有効です。コミュニティの外部にあるほかの仮想ネットワークからの直接のトラフィックは許可されません。

注記 - ターゲットサービスドメインが PVLAN 機能をサポートしない場合、PVLAN 用に構成されているゲストドメインの移行は失敗することがあります。

PVLAN の要件

`ldm add-vnet` コマンドと `ldm set-vnet` コマンドを使用すると PVLAN を構成できます。これらのコマンドを使用して `pvlan` プロパティを設定します。PVLAN を正しく構成するには `pvid` プロパティも指定する必要があります。

この機能には、少なくとも Oracle Solaris 11.2 SRU 4 OS が必要です。

PVLAN を構成するには、次の情報を指定する必要があります。

- **プライマリ VLAN ID。**プライマリ VLAN ID は、1 つの仮想ネットワークデバイスの PVLAN を構成するために使用されるポート VLAN ID (PVID) です。この構成により、ゲストドメインが VLAN パケットを確実に受信できるようになります。PVLAN の VID を構成することはできません。この値は `pvid` プロパティによって表されます。
- **セカンダリ VLAN ID。**セカンダリ VLAN ID は、PVLAN 機能を提供するために特定の VLAN によって使用されます。この情報は、`pvlan` 値の *secondary-vid* 部分として指定します。*secondary-vid* は、1 から 4094 の範囲の整数値です。プライマリ VLAN には多数のセカンダリ VLAN を含めることができますが、次の制限があります。
 - プライマリ VLAN ID もセカンダリ VLAN ID もデフォルトの VLAN ID と同じにすることはできません。
 - 隔離とコミュニティのどちらの PVLAN タイプでも、プライマリ VLAN ID とセカンダリ VLAN ID を同じ値にすることはできません。
 - 各プライマリ VLAN は、分離された PVLAN を 1 つだけ構成できます。したがって、同じプライマリ VLAN ID を使用する 2 つの分離された PVLAN を作成することはできません。
 - プライマリ VLAN には複数のコミュニティ VLAN を含めることができますが、次の制限があります。
 - プライマリ VLAN ID をセカンダリ VLAN ID として使用して別のコミュニティ PVLAN を作成することはできません。

たとえば、3 のプライマリ VLAN ID と 100 のセカンダリ VLAN ID を持つコミュニティ PVLAN がある場合は、3 をセカンダリ VLAN ID として使用する別のコミュニティ PVLAN を作成することはできません。

- セカンダリ VLAN ID をプライマリ VLAN ID として使用してコミュニティ PVLAN を作成することはできません。

たとえば、3 のプライマリ VLAN ID と 100 のセカンダリ VLAN ID を持つコミュニティ PVLAN がある場合は、100 をプライマリ VLAN ID として使用する別のコミュニティ PVLAN を作成することはできません。

- セカンダリ VLAN ID が、通常の仮想ネットワークまたは VNIC の VLAN ID としてすでに使用されているはいけません。



注意 - Logical Domains Manager は、特定の仮想スイッチ上の仮想ネットワークの構成しか検証できません。同じバックエンドデバイス上の Oracle Solaris VNIC に対して PVLAN 構成を設定する場合は、すべての VNIC および仮想ネットワークにわたって同じ要件が満たされていることを確認してください。

- **PVLAN タイプ**。この情報は `pvlan` 値の `pvlan-type` 部分として指定します。`pvlan-type` は、次の値のいずれかです。
 - **isolated**。隔離 PVLAN に関連付けられたポートは、バックエンドネットワークデバイス上のすべてのピア仮想ネットワークおよび Oracle Solaris 仮想 NIC から隔離されます。パケットは、その PVLAN に指定された値に基づく外部ネットワークにのみ到達します。
 - **community**。コミュニティ PVLAN に関連付けられたポートは、同じコミュニティ PVLAN 内に存在するほかのポートと通信できますが、その他のすべてのポートから隔離されます。パケットは、その PVLAN に指定された値に基づく外部ネットワークに到達します。

PVLAN の構成

このセクションには、PVLAN を作成し、PVLAN に関する情報を一覧表示する方法について説明するタスクが含まれています。

PVLAN の作成

PVLAN は、`ldm add-vnet` または `ldm set-vnet` コマンドを使用して `pvlan` プロパティ値を設定することによって構成できます。[ldm\(1M\)](#) マニュアルページを参照してください。

次のコマンドを使用すると、PVLAN を作成または削除できます。

- `ldm add-vnet` を使用して PVLAN を作成します。

```
ldm add-vnet pvid=port-VLAN-ID pvlan=secondary-vid,pvlan-type \
if-name vswitch-name domain-name
```

次のコマンドは、4 のプライマリ `vlan-id`、200 のセカンダリ `vlan-id`、および `isolated` の `pvlan-type` を持つ PVLAN で仮想ネットワークを作成する方法を示しています。

```
primary# ldm add-vnet pvid=4 pvlan=200,isolated vnet1 primary-vsw0 ldg1
```

- `ldm set-vnet` を使用して PVLAN を作成します。

```
ldm set-vnet pvid=port-VLAN-ID pvlan=secondary-vid,pvlan-type if-name domain-name
```

次のコマンドは、3 のプライマリ `vlan-id`、300 のセカンダリ `vlan-id`、および `community` の `pvlan-type` を持つ PVLAN で仮想ネットワークを作成する方法を示しています。

```
primary# ldm add-vnet pvid=3 pvlan=300,community vnet1 primary-vsw0 ldg1
```

- `ldm set-vnet` を使用して PVLAN を削除します。

```
ldm set-vnet pvlan= if-name vswitch-name domain-name
```

次のコマンドは、`vnet0` 仮想ネットワークの PVLAN 構成を削除します。このコマンドの結果として、指定された仮想ネットワークは、PVLAN を構成するときに指定した `vlan-id` を使用する通常の VLAN になります。

```
primary# ldm set-vnet pvlan= vnet0 primary-vsw0 ldg1
```

PVLAN 情報の表示

PVLAN に関する情報は、Logical Domains Manager のいくつかの一覧表示サブコマンドを使用して表示できます。[ldm\(1M\)](#) マニュアルページを参照してください。

次のコマンドを使用すると、PVLAN 情報を表示できます。

- `ldm list-domain -o network` を使用して PVLAN 情報を一覧表示します。

```
ldm list-domain [-e] [-l] -o network [-p] [domain-name...]
```

次の例は、ldm list-domain -o network コマンドを使用した場合の ldg1 ドメイン上の PVLAN 構成に関する情報を示しています。

- 次の ldm list-domain コマンドは、ldg1 ドメイン上の PVLAN 構成に関する情報を表示します。

```
primary# ldm list-domain -o network ldg1

NAME
ldg1

MAC
00:14:4f:fa:bf:0f

NETWORK
NAME      SERVICE          ID DEVICE    MAC
vnet0     primary-vsw0@primary 0  network@0  00:14:4f:f8:03:ed
MODE      PVID VID          MTU  MAXBW  LINKPROP
          1    3              1500 1700
PVLAN : 200,community
```

- 次の ldm list-domain コマンドは、ldg1 ドメインの PVLAN 構成情報を解析可能な形式で表示します。

```
primary# ldm list-domain -o network -p ldg1

VERSION 1.13
DOMAIN|name=ldg1|
MAC|mac-addr=00:14:4f:fa:bf:0f
VNET|name=vnet0|dev=network@0|service=primary-vsw0@primary
|mac-addr=00:14:4f:f8:03:ed|mode=|pvid=1|vid=3|mtu=1500|linkprop=|id=0
|alt-mac-addr=|maxbw=1700|protect=|priority=|cos=|pvlan=200,community
```

- ldm list-bindings を使用して PVLAN 情報を一覧表示します。

```
ldm list-bindings [-e] [-p] [domain-name...]
```


次の例は、`ldm list-bindingsnetwork` コマンドを使用した場合の `ldg1` ドメイン上の PVLAN 構成に関する情報を示しています。

- 次の `ldm list-bindings` コマンドは、`ldg1` ドメイン上の PVLAN 構成に関する情報を表示します。

```
primary# ldm list-bindings
...
NETWORK
NAME      SERVICE              ID DEVICE    MAC
vnet0     primary-vsw0@primary 0  network@0  00:14:4f:f8:03:ed
MODE      PVID VID    MTU  MAXBW LINKPROP
          1    3      1500 1700
          PVLAN :200,community
PEER              MAC              MODE PVID VID    MTU  MAXBW LINKPROP
primary-vsw0@primary 00:14:4f:f8:fe:5e 1
```

- 次の `ldm list-bindings` コマンドは、`ldg1` ドメインの PVLAN 構成情報を解析可能な形式で表示します。

```
primary# ldm list-bindings -p
...
VNET|name=vnet0|dev=network@0|service=primary-vsw0@primary
|mac-addr=00:14:4f:f8:03:ed|mode=|pvid=1|vid=3|mtu=1500|linkprop=
|id=0|alt-mac-addr=|maxbw=1700|protect=|priority=|cos=|pvlan=200,community
|peer=primary-vsw0@primary|mac-addr=00:14:4f:f8:fe:5e|mode=|pvid=1|vid=
|mtu=1500|maxbw=
```

- `ldm list-constraints` を使用して PVLAN 情報を一覧表示します。

```
ldm list-constraints [-x] [domain-name...]
```

次に、`ldm list-constraints` コマンドを実行することによって生成された出力を示します。

```
primary# ldm list-constraints -x ldg1
...
<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>network</rasd:OtherResourceType>
    <rasd:Address>auto-allocated</rasd:Address>
```

```

<gprop:GenericProperty key="vnet_name">vnet0</gprop:GenericProperty>
<gprop:GenericProperty key="service_name">primary-vsw0</gprop:GenericProperty>
<gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
<gprop:GenericProperty key="vid">3</gprop:GenericProperty>
<gprop:GenericProperty key="pvlan">200,community</gprop:GenericProperty>
<gprop:GenericProperty key="maxbw">1700000000</gprop:GenericProperty>
<gprop:GenericProperty key="device">network0</gprop:GenericProperty>
<gprop:GenericProperty key="id">0</gprop:GenericProperty>
</Item>

```

パケットスルーブットパフォーマンスの調整

ldm add-vnet コマンドと ldm set-vnet コマンドを使用すると、パケットスルーブットパフォーマンスを調整するための次のデータリンクプロパティ値を設定できます。

priority	CPU パケット処理の優先度を指定します
cos	リンクの IEEE 802.1p リンクサービスクラスを指定します
protection	パケットトラフィックセキュリティの種類を指定します

有効およびデフォルトのプロパティ値については、[ldm\(1M\)](#)のマニュアルページを参照してください。

例 12-9 データリンクパケットプロパティの設定および表示

次のサンプルは、ldm set-vnet コマンドを使用して、priority、protection、および cos プロパティ値を 1 つのコマンドで設定する方法を示します。ldm add-vnet コマンドを使用して、指定されたデータリンクプロパティ値を使用する新しい仮想ネットワークを追加することもできます。

```

primary# ldm set-vnet allowed-ips=192.168.100,1,192.168.100.2 \
allowed-dhpcids=oracle1@system1.company.com, \
00:14:4f:f9:d3:88,system2,00:14:4f:fb:61:6e cos=7 priority=high \
protection=restricted,mac-nospoof,ip-nospoof,dhcp-nospoof vnet3_ldg3 ldg3

```

ldm list -o network コマンドは、前の ldm set-vnet コマンドで設定した ldg3 ドメイン上のデータリンクプロパティ値を表示します。protection の値は、mac-nospoof、restricted、192.168.100,1,192.168.100.2 MAC アドレスの場合の ip-nospoof、および system1@company.com,00:14:4f:f9:d3:88,system2,00:14:4f:fb:61:6e の場合の

dhcp-nospoofです。priority は high に設定され、サービスのクラス (cos) は 7 に設定されます。

```
primary# ldm list -o network ldg3
NAME
ldg3

MAC
00:14:4f:fa:3b:49

VSW
NAME      MAC      NET-DEV ID DEVICE  LINKPROP DEFAULT-VLAN-ID
PVID VID  MTU  MODE INTER-VNET-LINK
ldg3_vsw1 00:14:4f:f9:65:b5 net3    0  switch@0 1      1
          1500      on

NETWORK
NAME      SERVICE      ID DEVICE  MAC      MODE PVID VID
MTU  MAXBW LINKPROP
vnet3      primary-vsw0@primary 0  network@0 00:14:4f:fa:47:f4      1
1500
vnet1_ldg3 primary-vsw1@primary 1  network@1 00:14:4f:fb:90:b7      1
1500
vnet2_ldg3 ldg1_vsw1@ldg1      2  network@2 00:14:4f:fb:61:6e      1
1500
vnet3_ldg3 ldg1_vsw1@ldg1      3  network@3 00:14:4f:fb:25:70      1
                                00:14:4f:f8:7a:8d
                                00:14:4f:fb:6e:32
                                00:14:4f:fb:1a:2f

1500 1000
PROTECTION: mac-nospoof
            restricted
            ip-nospoof/192.168.100,1,192.168.100.2
            dhcp-nospoof/system1@oracle.us.oracle.com,
            00:14:4f:f9:d3:88,system2,00:14:4f:fb:61:6e
PRIORITY   : high
COS        : 7
```

NIU ハイブリッド I/O の使用

仮想 I/O フレームワークは、機能およびパフォーマンスを向上させるために、「ハイブリッド」I/O モデルを実装しています。ハイブリッド I/O モデルでは、ダイレクト I/O および仮想化 I/O を組み合わせることで、仮想マシンへの柔軟な I/O リソース配備が可能です。これは、仮想マシンに対して直接 I/O の機能が十分に提供されない場合、またはリソースの可用性や仮想マシンの移行のため、仮想マシンが持続的にあるいは一貫して直接 I/O を利用できない場合に特に便利です。

ハイブリッド I/O アーキテクチャーは、Oracle Sun UltraSPARC T2、SPARC T3、および SPARC T4 プラットフォームのネットワークインタフェースユニット (NIU) に適しています。NIU は、チップに統合されるネットワーク I/O インタフェースです。このアーキテクチャーにより、ダイレクトメモリアccess (DMA) リソースを仮想ネットワークデバイスに動的に割り当てることができ、ドメイン内のアプリケーションのパフォーマンスが安定します。

注記 - SR-IOV が使われるようになって NIU Hybrid I/O 機能は非推奨になりました。Oracle VM Server for SPARC 3.3 は、この機能を含む最後のソフトウェアリリースです。

NIU ハイブリッド I/O は、Oracle Sun UltraSPARC T2、SPARC T3、および SPARC T4 プラットフォームで使用できます。この機能は、仮想ネットワーク (vnet) デバイスに提供されるオプションのハイブリッドモードによって有効になります。このモードでは、DMA ハードウェアリソースが、パフォーマンスを向上させるために、ゲストドメインの vnet デバイ스에貸し出されます。ハイブリッドモードでは、ゲストドメインの vnet デバイスは、この DMA ハードウェアリソースを使用して、外部ネットワークとゲストドメインの間で、ユニキャストトラフィックを直接送受信することができます。同じシステム内のほかのゲストドメインへのブロードキャストトラフィック、マルチキャストトラフィック、およびユニキャストトラフィックは、仮想 I/O 通信メカニズムを使用して引き続き送信されます。

注記 - NIU ハイブリッド I/O は、UltraSPARC T2 Plus プラットフォームでは使用できません。

[図12-13「ハイブリッド仮想ネットワーク接続 \(Oracle Solaris 11\)」](#)と[図12-14「ハイブリッド仮想ネットワーク接続 \(Oracle Solaris 10\)」](#)はそれぞれ、Oracle Solaris 11 と Oracle Solaris 10 のハイブリッド I/O 構成を示しています。

図 12-13 ハイブリッド仮想ネットワーク接続 (Oracle Solaris 11)

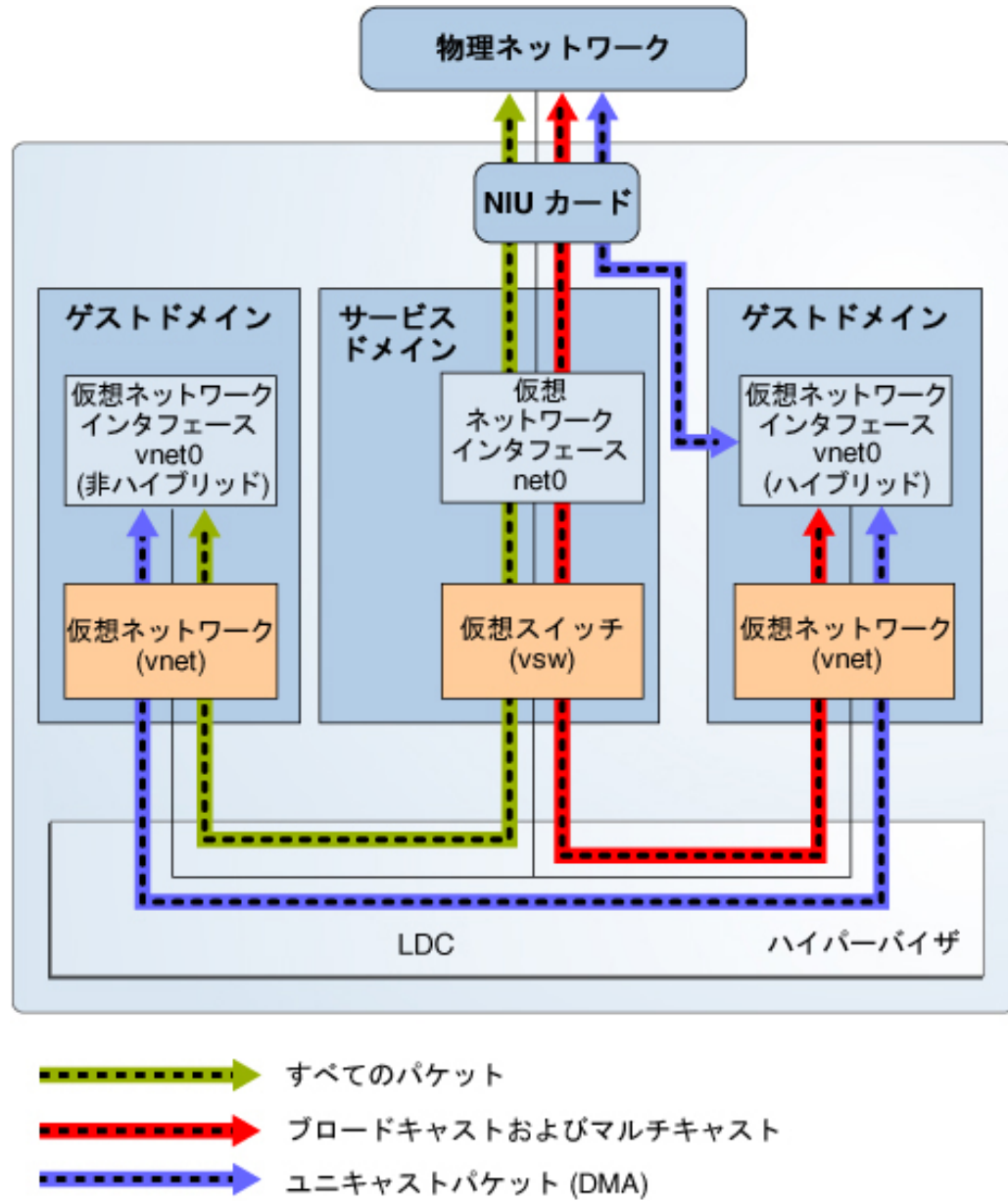
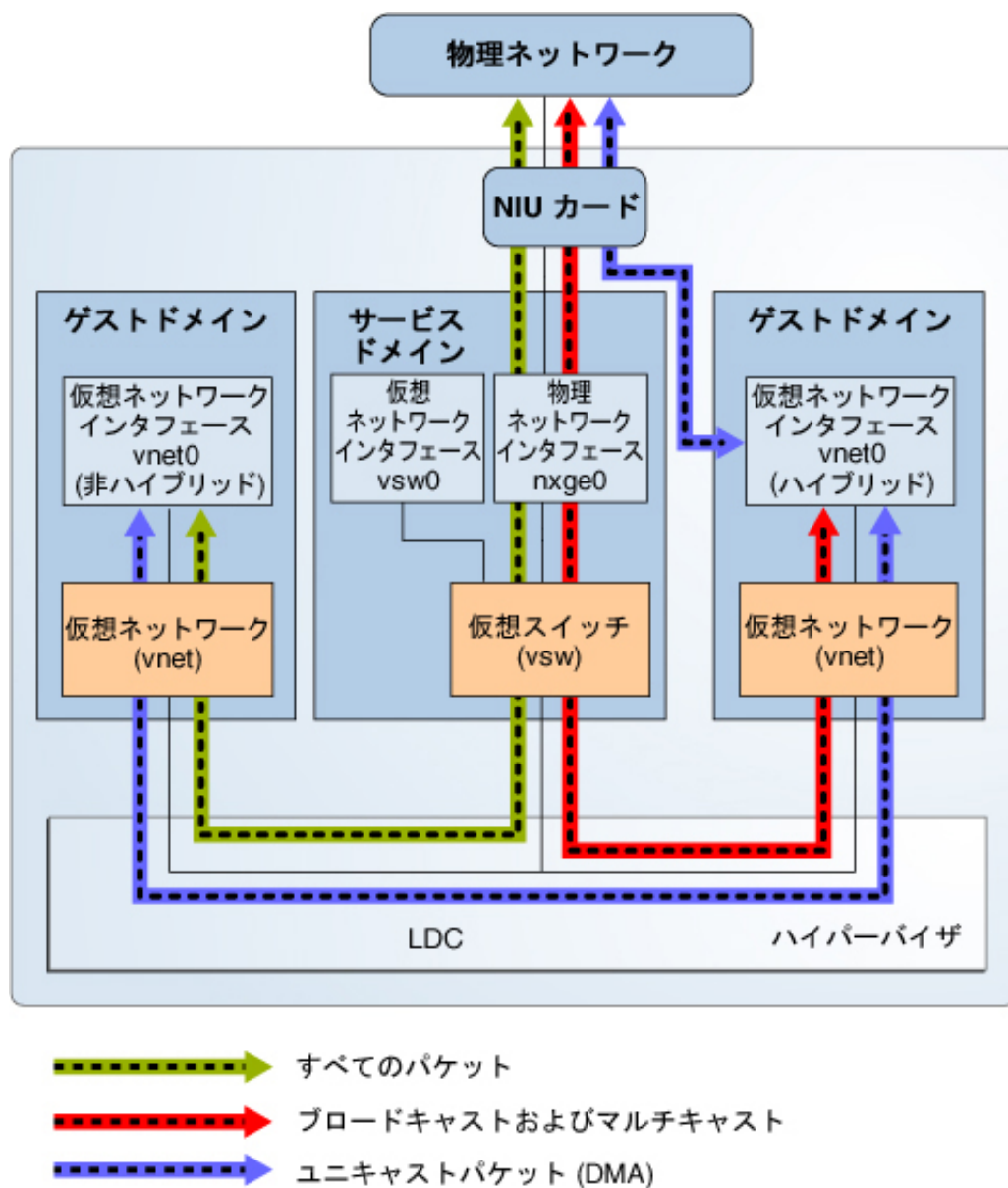


図 12-14 ハイブリッド仮想ネットワーク接続 (Oracle Solaris 10)



ハイブリッドモードは、NIU ネットワークデバイスを使用するように構成された仮想スイッチ (vsw) に関連付けられた vnet デバイスだけに適用されます。共有可能な DMA ハードウェアリ

ソースには制限があるため、DMA ハードウェアリソースの割り当てを受けられるのは、一度に、1 つの `vsw` あたり最大 3 つの `vnet` デバイスのみです。4 つ以上の `vnet` デバイスでハイブリッドモードを有効にすると、割り当ては先着順に行われます。1 つのシステムに 2 つの NIU ネットワークデバイスがあるため、DMA ハードウェアリソースが割り当てられている 2 つの異なる仮想スイッチで、合計 6 つの `vnet` デバイスが存在できます。

この機能を使用する場合は、次の点に注意してください。

- `vnet` デバイスのハイブリッドモードオプションは、推奨としてのみ扱われるため、DMA リソースが割り当てられるのは、DMA リソースが利用可能で、デバイスがそれらを使用できる場合だけです。
- Logical Domains Manager CLI コマンドは、ハイブリッドモードオプションを検証しません。つまり、どの `vnet` にも、いくつかの `vnet` デバイスにもハイブリッドモードを設定することができます。
- ゲストドメインおよびサービスドメインでは、Oracle Solaris 10 10/08 以上の OS を実行する必要があります。
- DMA ハードウェアリソースの貸し出しを受けられるのは、一度に、1 つの `vsw` あたり最大 3 つの `vnet` デバイスのみです。2 つの NIU ネットワークデバイスがあるため、DMA ハードウェアリソースの貸し出しを受けられるのは合計 6 つの `vnet` デバイスです。

注記 - 1 つの `vsw` あたり 3 つの `vnet` デバイスのみにハイブリッドモードを設定して、DMA ハードウェアリソースが確実に割り当てられるようにしてください。

- ゲストドメインがアクティブの間、ハイブリッドモードオプションを動的に変更することはできません。
- DMA ハードウェアリソースが割り当てられるのは、ゲストドメインでアクティブな `vnet` デバイスが作成されている場合のみです。
- NIU 10 ギガビット Ethernet ドライバ (`nxge`) は NIU カードで使用されています。他の 10 ギガビットネットワークカードでも、同じドライバが使用されます。ただし、NIU ハイブリッド I/O 機能は、NIU ネットワークデバイスのみで利用可能です。

▼ NIU ネットワークデバイスで仮想スイッチを構成する方法

1. NIU ネットワークデバイスを決定します。

次の例は、UltraSPARC T2 サーバーの出力を示しています。

```
primary# grep nxge /etc/path_to_inst
"/niu@80/network@0" 0 "nxge"
"/niu@80/network@1" 1 "nxge"
```

次の例は、SPARC T3-1 または SPARC T4-1 サーバーの出力を示しています。

```
primary# grep nxge /etc/path_to_inst
"/niu@480/network@0" 0 "nxge"
"/niu@480/network@1" 1 "nxge"
```

2. Oracle Solaris 11 OS のみ: nxge0 などの NIU ネットワークデバイスに対応するリンク名を識別します。

```
primary# dladm show-phys -L |grep nxge0
net2                nxge0                /SYS/MB
```

3. 仮想スイッチを構成します。

次の例では nxge0 の代わりに net2 を使用しています。

```
primary# ldm add-vsw net-dev=net2 primary-vsw0 primary
```

▼ ハイブリッドモードを有効または無効にする方法

vnet デバイスでは、ハイブリッドモードがデフォルトで無効になっているため、明示的に有効にする必要があります。

- ハイブリッドモードを有効または無効にするには、ldm コマンドを使用します。

- 作成中に vnet デバイスのハイブリッドモードを有効にするには:

```
primary# ldm add-vnet mode=hybrid vnet01 primary-vsw0 ldom01
```

- vnet デバイスのハイブリッドモードを無効にするには:

```
primary# ldm set-vnet mode= vnet01 ldom01
```

仮想スイッチでのリンクアグリゲーションの使用

リンクアグリゲーションを使用するように仮想スイッチを構成できます。リンクアグリゲーションは、物理ネットワークに接続するための仮想スイッチのネットワークデバイスとして使用します。こ

の構成を使用すると、仮想スイッチで IEEE 802.3ad Link Aggregation Standard によって提供される機能を利用できます。この機能には、帯域幅の増加、負荷分散、フェイルオーバーなどが含まれます。リンクアグリゲーションを構成する方法については、『[Managing Network Datalinks in Oracle Solaris 11.3](#)』の「[Creating a Link Aggregation](#)」を参照してください。

リンクアグリゲーションを作成したら、そのリンクアグリゲーションを仮想スイッチに割り当てることができます。この割り当て方法は、仮想スイッチへの物理ネットワークデバイスの割り当てに似ています。ldm add-vswitch または ldm set-vswitch コマンドを使用して net-dev プロパティを設定します。

リンクアグリゲーションを仮想スイッチに割り当てると、物理ネットワークに対して送受信されるトラフィックは集積体を通過しています。必要な負荷分散またはフェイルオーバーは、ベースとなる集積体のフレームワークによって透過的に処理されます。リンクアグリゲーションは、ゲストドメイン上の仮想ネットワーク (vnet) デバイスに対して、および集積体を使用する仮想スイッチにバインドされた仮想ネットワークデバイスに対して、完全に透過的です。

注記 - 仮想ネットワークデバイス (vnet および vsw) をリンクアグリゲーションにグループ化することはできません。

サービスドメインでリンクアグリゲーションを使うように構成された仮想スイッチを、作成して使用できます。[47 ページの「仮想スイッチをプライマリインタフェースとして構成する方法」](#)を参照してください。

図12-15「リンクアグリゲーションを使用するための仮想スイッチの構成 (Oracle Solaris 11)」と図12-16「リンクアグリゲーションを使用するための仮想スイッチの構成 (Oracle Solaris 10)」はそれぞれ、物理インタフェース net0 および net1 と nxge0 および nxge1 経由でアグリゲーション (aggr1) を使用するように構成された仮想スイッチを示しています。

図 12-15 リンクアグリゲーションを使用するための仮想スイッチの構成 (Oracle Solaris 11)

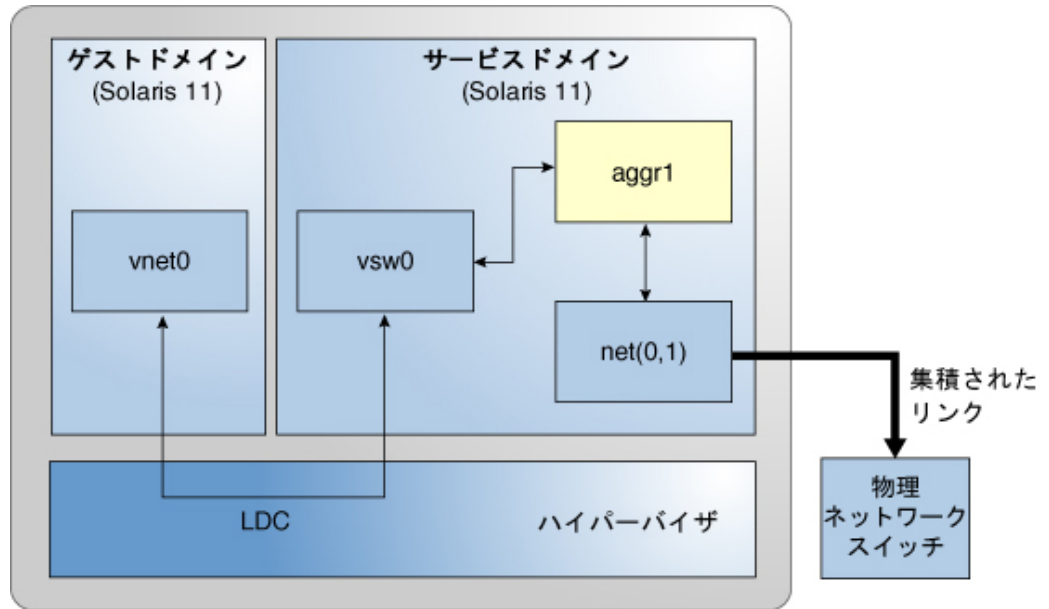
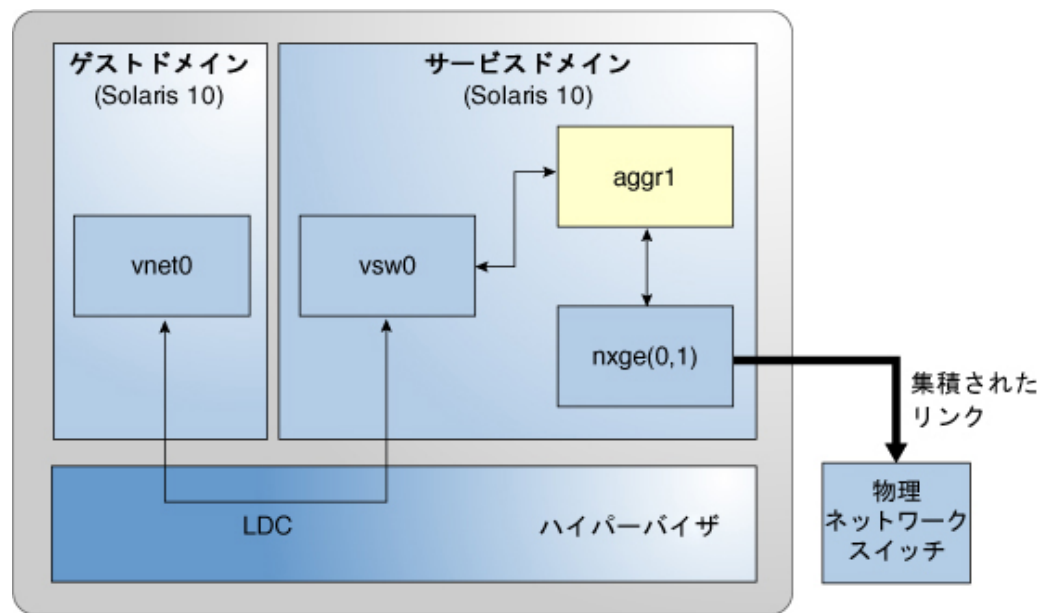


図 12-16 リンクアグリゲーションを使用するための仮想スイッチの構成 (Oracle Solaris 10)



ジャンボフレームの構成

Oracle VM Server for SPARC の仮想スイッチ (vsw) および仮想ネットワーク (vnet) デバイスで、1500 バイトを超えるペイロードサイズの Ethernet フレームをサポートできるようになりました。そのため、これらのドライバはネットワークのスループットを向上できるようになりました。

ジャンボフレームを有効にするには、仮想スイッチデバイスの最大転送単位 (MTU) を指定します。このような場合、仮想スイッチデバイスとその仮想スイッチデバイスにバインドされているすべての仮想ネットワークデバイスで、指定した MTU 値が使用されます。

仮想ネットワークデバイスに必要な MTU 値が仮想スイッチによってサポートされる MTU 値よりも小さい場合、仮想ネットワークデバイスに直接 MTU 値を指定できます。

注記 - Oracle Solaris 10 5/09 OS でのみ、物理デバイスの MTU を仮想スイッチの MTU に一致するように構成する必要があります。特定のドライバの構成については、Oracle Solaris リファレンスマニュアルのセクション 7D にある、そのドライバに対応するマニュアルページを参照してください。たとえば、Oracle Solaris 10 `nxge` ドライバの情報については、[nxge\(7D\)](#) のマニュアルページを参照してください。

まれに、`ldm add-vnet` または `ldm set-vnet` コマンドを使用して、仮想スイッチの MTU 値と異なる MTU 値を仮想ネットワークデバイスに指定する必要がある場合があります。たとえば、VLAN を仮想ネットワークデバイス上で構成し、VLAN の MTU の最大値が仮想スイッチの MTU 値よりも小さい場合、仮想ネットワークデバイスの MTU 値を変更する場合があります。デフォルトの MTU 値のみが使用されているドメインでは、ジャンボフレームをサポートしている `vnet` ドライバは必要ない場合があります。ただし、ジャンボフレームを使用する仮想スイッチにバインドされた仮想ネットワークデバイスがドメインに存在する場合、`vnet` ドライバがジャンボフレームをサポートしていることを確認してください。

`ldm set-vnet` コマンドを使用して仮想ネットワークデバイスで `mtu` 値を指定する場合、あとで仮想スイッチデバイスの MTU 値が更新されても、仮想ネットワークデバイスには更新値は伝播されません。仮想ネットワークデバイスを再度有効にして仮想スイッチデバイスから MTU 値を取得するには、次のコマンドを実行します。

```
primary# ldm set-vnet mtu= vnet-name domain-name
```

仮想ネットワークデバイスでジャンボフレームを有効にすると、その仮想ネットワークデバイスに割り当てられているハイブリッド I/O リソースでもジャンボフレームが自動的に有効になります。

制御ドメインでは、Logical Domains Manager が、`ldm set-vsw` および `ldm set-vnet` コマンドによって設定された MTU 値を遅延再構成処理として更新します。制御ドメイン以外のドメインの MTU を更新するには、ドメインを停止してから `ldm set-vsw` または `ldm set-vnet` コマンドを実行して MTU 値を変更する必要があります。

▼ ジャンボフレームを使用するように仮想ネットワークおよび仮想スイッチデバイスを構成する方法

1. 制御ドメインにログインします。
2. 管理者になります。

Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「About Using Rights to Control Users and Processes」を参照してください。

3. 仮想ネットワークで使用する MTU の値を決定します。

1500 - 16000 バイトの MTU 値を指定できます。指定する MTU は、仮想スイッチに割り当てられた物理ネットワークデバイスの MTU と一致する必要があります。

4. 仮想スイッチデバイスまたは仮想ネットワークデバイスの MTU 値を指定します。

次のいずれかの手順を実行します。

- MTU を `mtu` プロパティの値として指定することで、サービストメインの新しい仮想スイッチデバイスでジャンボフレームを有効にします。

```
primary# ldm add-vsw net-dev=device mtu=value vswitch-name ldom
```

このコマンドは、仮想スイッチの構成に加えて、この仮想スイッチにバインドされる各仮想ネットワークデバイスの MTU 値を更新します。

- MTU を `mtu` プロパティの値として指定することで、サービストメインの既存の仮想スイッチデバイスでジャンボフレームを有効にします。

```
primary# ldm set-vsw net-dev=device mtu=value vswitch-name
```

このコマンドは、仮想スイッチの構成に加えて、この仮想スイッチにバインドされる各仮想ネットワークデバイスの MTU 値を更新します。

例 12-10 仮想スイッチおよび仮想ネットワークデバイスでのジャンボフレームの構成

- 次の例に、MTU 値が 9000 の新しい仮想スイッチデバイスを追加する方法を示します。この MTU 値は、仮想スイッチデバイスからすべてのクライアントの仮想ネットワークデバイスに伝播されます。

まず、`ldm add-vsw` コマンドを使用して、MTU 値 9000 で仮想スイッチデバイス `ldg1-vsw0` を作成します。ネットワークデバイス `net0` のインスタンス 0 が `net-dev` プロパティの値として指定されていることに注意してください。

```
primary# ldm add-vsw net-dev=net0 mtu=9000 ldg1-vsw0 ldg1
```

次に、`ldm add-vnet` コマンドを使用して、この仮想スイッチ `ldg1-vsw0` にクライアントの仮想ネットワークデバイスを追加します。仮想ネットワークデバイスの MTU は、バインドされて

いる仮想スイッチから暗黙に割り当てられます。そのため、`ldm add-vnet` コマンドで `mtu` プロパティの値を指定する必要はありません。

```
primary# ldm add-vnet vnet01 ldg1-vsw0 ldg1
```

実行している Oracle Solaris OS のバージョンに応じて、次を実行します。

- **Oracle Solaris 11 OS:** プライマリインタフェースの `mtu` プロパティ値を表示するには、`ipadm` コマンドを使用します。

```
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu      ipv4  rw   9000    --        9000    68-9000
```

`ipadm` コマンドを使用して、ゲストドメイン `ldg1` 内に仮想ネットワークインタフェースを作成します。`ipadm show-ifprop` コマンドの出力には、`mtu` プロパティの値が `9000` であることが示されます。

```
ldg1# ipadm create-ip net0
ldg1# ipadm create-addr -T static -a 192.168.1.101/24 net0/ipv4
ldg1# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu      ipv4  rw   9000    --        9000    68-9000
```

- **Oracle Solaris 10 OS:** `ifconfig` コマンドを使用して、サービスドメイン `ldg1` 内に仮想スイッチインタフェースを作成します。`ifconfig vsw0` コマンドの出力には、`mtu` プロパティの値が `9000` であることが示されます。

```
ldg1# ifconfig vsw0 plumb
ldg1# ifconfig vsw0 192.168.1.100/24 up
ldg1# ifconfig vsw0
vsw0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 5
      inet 192.168.1.100 netmask ffffffff broadcast 192.168.1.255
      ether 0:14:4f:fa:0:99
```

`ifconfig` コマンドを使用して、ゲストドメイン `ldg1` 内に仮想ネットワークインタフェースを作成します。`ifconfig vnet0` コマンドの出力には、`mtu` プロパティの値が `9000` であることが示されます。

```
ldg1# ifconfig vnet0 plumb
ldg1# ifconfig vnet0 192.168.1.101/24 up
```

```
ldg1# ifconfig vnet0
vnet0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 4
        inet 192.168.1.101 netmask ffffffff00 broadcast 192.168.1.255
        ether 0:14:4f:f9:c4:13
```

- 次の例に、インタフェースの MTU を 4000 に変更する方法を示します。

インタフェースの MTU は、Logical Domains Manager によってデバイスに割り当てられた MTU よりも小さい値にのみ変更できます。この方法は、VLAN が構成されていて各 VLAN インタフェースに異なる MTU が必要なときに便利です。

- **Oracle Solaris 11 OS:** ipadm コマンドを使用します。

```
primary# ipadm set-ifprop -p mtu=4000 net0
primary# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu      ipv4  rw   4000    --         9000    68-9000
```

- **Oracle Solaris 10 OS:** ifconfig コマンドを使用します。

```
primary# ifconfig vnet0 mtu 4000
primary# ifconfig vnet0
vnet0: flags=1201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS,FIXEDMTU>
mtu 4000 index 4
        inet 192.168.1.101 netmask ffffffff00 broadcast 192.168.1.255
        ether 0:14:4f:f9:c4:13
```

ジャンボフレームに対応していない旧バージョンの vnet および vsw ドライバとの互換性 (Oracle Solaris 10)

注記 - このセクションは Oracle Solaris 10 OS にのみ適用されます。

ジャンボフレームをサポートしているドライバとジャンボフレームをサポートしていないドライバを、同じシステム上で相互運用できます。この相互運用は、仮想スイッチを作成するときにジャンボフレームのサポートが有効になっていない場合にかぎり可能です。

注記 - 仮想スイッチに関連付けられたゲストドメインまたはサービسدメインがジャンボフレームをサポートしている Oracle VM Server for SPARC ドライバを使用していない場合、mtu プロパティを設定しないでください。

ジャンボフレームを有効にするには、仮想スイッチの `mtu` プロパティをデフォルト値の 1500 から変更します。この場合、旧バージョンのドライバは `mtu` 設定を無視し、デフォルト値を引き続き使用します。`ldm list` の出力には、デフォルト値ではなく、指定した MTU 値が示されます。デフォルトの MTU よりも大きいフレームはそれらのデバイスには送られず、新しいドライバによって破棄されます。この場合、旧ドライバを使用し続けているゲストがあると、一貫性のないネットワーク動作につながる場合があります。この制限は、クライアントゲストドメインおよびサービスドメインの両方に当てはまります。

そのため、ジャンボフレームが有効な場合は、Oracle VM Server for SPARC ネットワークのすべての仮想デバイスをアップグレードし、ジャンボフレームをサポートしている新しいドライバが使用されるようにしてください。ジャンボフレームを構成するには、Logical Domains 1.2 以上を実行している必要があります。

Oracle Solaris 11 のネットワーク固有の機能の相違点

ドメインが Oracle Solaris 10 OS を実行している場合、Oracle VM Server for SPARC の一部のネットワーク機能の動作が Oracle Solaris 11 OS と比べて異なります。ドメインで Oracle Solaris 11 OS が実行されている場合の Oracle VM Server for SPARC 仮想ネットワークデバイスおよび仮想スイッチの機能の違いは次のようになります。

- **vswn** デバイスをプライマリネットワークインタフェースとして構成し、サービスドメインがゲストドメインと通信できるようにする

この構成は、Oracle Solaris 10 OS を実行するドメインにのみ必要です。Oracle Solaris 11 の場合、仮想スイッチは Oracle Solaris 11 ネットワークスタックを使用し、自動的にその仮想ネットワークデバイスが、`net0` などのそのバックエンドデバイスに対応するネットワークインタフェースと通信できるようにします。Oracle Solaris 11 では、`vswn` デバイスのネットワークインタフェースとしての構成はサポートされていません。

- **Oracle Solaris 11 etherstub** デバイスをバックエンドデバイスとして使用して、プライベート仮想スイッチを作成する

バックエンドデバイスに接続されていない場合、仮想スイッチは、ゲストドメイン間の通信のみを提供し、ゲストドメインとサービスドメインとの間の通信は提供しません。`etherstub` をバックエンドデバイスとして使用すると、ゲストドメインは Oracle Solaris 11 サービスドメインで構成されているゾーン (大域ゾーンを含む) と通信できます。この構成は、その `etherstub` に接続された VNIC を使用して実現されます。

- 仮想スイッチと仮想ネットワークデバイスに汎用名を使用する

Oracle Solaris 11 OS は `vsw` デバイスと `vnet` デバイスに汎用名を割り当てます。別の `vsw` デバイスまたは `vnet` デバイスであるバックエンドデバイスによって仮想スイッチを作成しないでください。汎用ネットワークデバイス名に関連付けられている実際の物理デバイスを表示するには、`dladm show-phys` コマンドを使用します。

■ Oracle Solaris 11 VNIC を使用して Ethernet スタブ上に VLAN を作成する

Oracle Solaris 11 サービスドメインの仮想スイッチインタフェース上に VLAN を構成しないでください。この構成はサポートされていません。代わりに、仮想スイッチの `net-dev` プロパティ値に対応するインタフェース上に VLAN を作成します。

Oracle Solaris 10 の場合は、ルーティングされる仮想スイッチを作成するために `null` 値を持つ `net-dev` プロパティを設定できます。ただし、Oracle Solaris 11 ではこの方法はサポートされていません。代わりに、Ethernet スタブデバイス上に VNIC を構成して VLAN の一部になるようにします。

次の例は、Ethernet スタブ上に VNIC を作成する方法を示しています。`dladm create-etherstub` コマンドは、Ethernet スタブ `estub100` を作成します。これは、仮想スイッチを作成するために `ldm add-vsw` コマンドによって使用される補助デバイスです。`ldm add-vsw` コマンドは、仮想スイッチを作成します。`dladm create-vnic` コマンドは、その仮想スイッチの VLAN を作成するために `etherstub` 上に VNIC を作成します。

```
primary# dladm create-etherstub estub100
primary# ldm add-vsw net-dev=estub100 vid=100 inter-vnet-link=off \
primary-vsw100 primary
primary# dladm create-vnic -l estub100 -m auto -v 100 vnic100
```

次の `ldm add-vnet` コマンドは、`ldg1` および `ldg2` ドメインが VLAN 100 経由で通信できるようにするための 2 つの VNIC を作成します。

```
primary# ldm add-vnet vid=100 ldg1-vnet100 primary-vsw100 ldg1
primary# ldm add-vnet vid=100 ldg2-vnet100 primary-vsw100 ldg2
```

次の例では、`dladm` コマンドは、`ldg1` および `ldg2` ゲストドメイン上に VLAN を作成します。`ipadm` コマンドは、`ldg1` および `ldg2` ドメイン上に作成した VNIC の IP アドレスを作成します。

```
ldg1# dladm create-vlan -l net1 -v 100 vlan100
ldg1# ipadm create-ip vlan100
ldg1# ipadm create-ipaddr -T static -a 192.168.100.10/24 vlan100/v4
ldg2# dladm create-vlan -l net1 -v 100 vlan100
```

```
ldg2# ipadm create-ip vlan100
ldg2# ipadm create-ipaddr -T static -a 192.168.100.20/24 vlan100/v4
```

■ 仮想スイッチと仮想ネットワークデバイスに汎用名を使用する

Oracle Solaris 11 OS は `vsw` デバイスと `vnet` デバイスに汎用名を割り当てます。別の `vsw` デバイスまたは `vnet` デバイスであるバックエンドデバイスによって仮想スイッチを作成しないでください。汎用ネットワークデバイス名に関連付けられている実際の物理デバイスを表示するには、`dladm show-phys` コマンドを使用します。

■ 仮想スイッチおよび仮想ネットワークデバイスで VNIC を使用する

`vsw` デバイスでは、VNIC を使用できません。`vsw` で VNIC を作成しようとしても失敗します。『Oracle VM Server for SPARC 3.3 リリースノート』の「Oracle Solaris 11: 自動ネットワークインタフェースで構成されたゾーンの起動に失敗することがある」を参照してください。

■ Oracle Solaris 11 ゲストドメイン上でネットワーク可観測性コマンドを使用する

`ldm list-netdev` および `ldm list-netstat` コマンドを使用して、Oracle Solaris 11 ゲストドメインに関する情報を取得できます。

仮想ネットワーク上の仮想 NIC の使用

Oracle Solaris 11 OS を使用すると、仮想ネットワークインタフェースカード (VNIC)、仮想スイッチ、および `etherstub` で構成される仮想ネットワークを定義できます。Oracle Solaris ゾーンはオペレーティングシステムサービスを仮想化し、ある論理ドメインの同じ Oracle Solaris OS インスタンス内で実行中の複数のアプリケーションのために、隔離されたセキュアな複数の環境を提供します。

Oracle Solaris 11 では、各ゾーンが大域ゾーンからネットワークプロパティを継承し、独自のネットワークアドレスやその他のプロパティを設定できない Oracle Solaris 10 の「共有 IP」ゾーンモデルが改良されています。現在では、仮想ネットワークデバイスを含むゾーンを使用して、複数の分離された仮想 NIC を構成したり、ゾーンを各仮想ネットワークに関連付けたり、分離、接続、およびサービスの品質 (QoS) のための規則を確立したりできます。

詳細は、Oracle Solaris 11.3 Information Library (http://docs.oracle.com/cd/E53394_01/) にあるネットワークに関する本を参照してください。

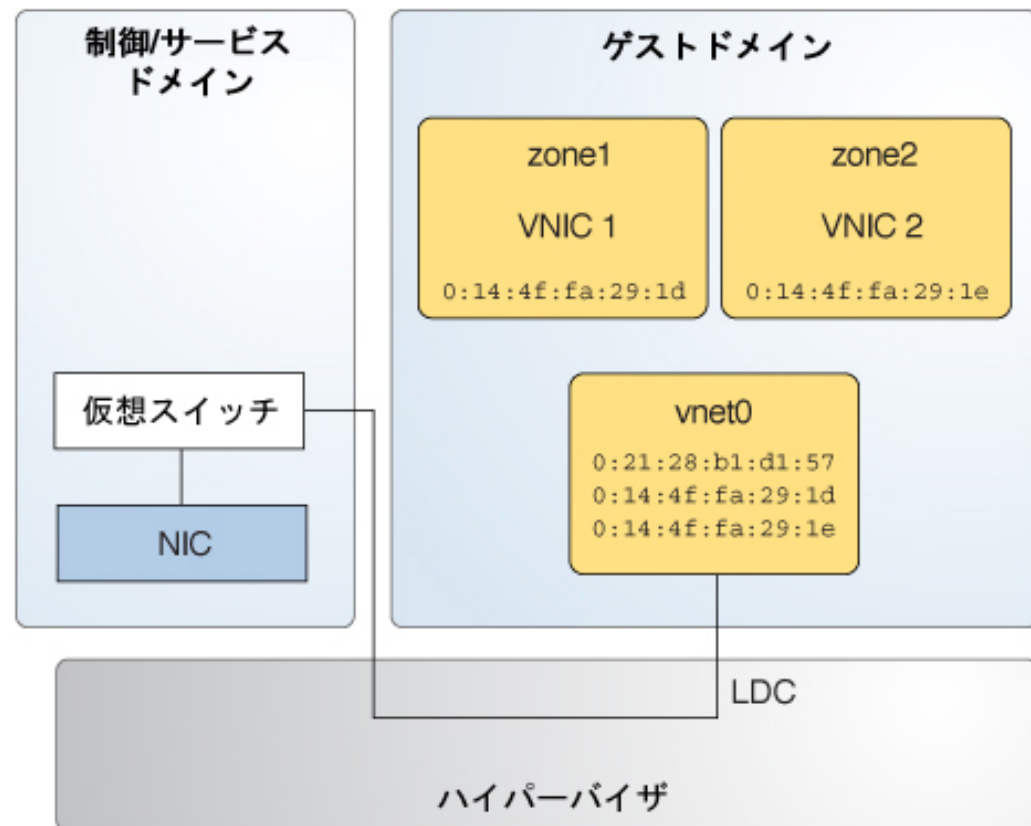
論理ドメイン内の仮想ネットワークデバイスは、複数の Oracle Solaris 11 仮想 NIC をサポートできます。仮想ネットワークデバイスは、それがサポートする仮想 NIC ごとに 1 つずつの複

数の MAC アドレスをサポートするように構成されている必要があります。論理ドメイン内の Oracle Solaris ゾーンは仮想 NIC に接続します。

図12-17「仮想ネットワークデバイス上の仮想 NIC」は、Oracle Solaris OS に `vnet1` という名前の 1 つの仮想ネットワークデバイスを提供する論理ドメイン `domain1` を示しています。この仮想ネットワークデバイスは、それぞれ独自の MAC アドレスを持ち、ゾーンに個別に割り当て可能な複数の Oracle Solaris 11 仮想ネットワークデバイスをホストできます。

`domain1` ドメイン内には、`zone1`、`zone2`、`zone3` という 3 つの Oracle Solaris 11 ゾーンがあります。各ゾーンは、`vnet1` 仮想ネットワークデバイスに基づいて、仮想 NIC によってネットワークに接続されます。

図 12-17 仮想ネットワークデバイス上の仮想 NIC



次のセクションでは、仮想ネットワークデバイス上の仮想 NIC の構成、および仮想 NIC によるドメイン内のゾーンの作成について説明します。

- [308 ページの「仮想ネットワークデバイス上の仮想 NIC の構成」](#)
- [309 ページの「ドメイン内での Oracle Solaris 11 ゾーンの作成」](#)

Ethernet SR-IOV 仮想機能での仮想 NIC の使用については、次のセクションを参照してください。

- [88 ページの「Ethernet 仮想機能の作成」](#)
- [96 ページの「Ethernet SR-IOV 仮想機能の変更」](#)
- [102 ページの「SR-IOV 仮想機能での VNIC の作成」](#)

仮想ネットワークデバイス上の仮想 NIC の構成

仮想ネットワークデバイス上の仮想 NIC を構成するには、制御ドメインは Oracle Solaris 11.1 SRU 4 OS 以上を実行し、ゲストドメインは Oracle Solaris 11.1 OS 以上を実行している必要があります。

複数の MAC アドレスをホストするように仮想ネットワークデバイスを構成するには、`ldm add-vnet` または `ldm set-vnet` コマンドを使用して、`alt-mac-addr` プロパティにコンマで区切られた 1 つ以上の値を指定します。有効な値は、オクテット MAC アドレスおよび `auto` です。`auto` 値は、システムが MAC アドレスを生成することを示します。

たとえば、次の方法のいずれかで、仮想ネットワークデバイス用にシステムで生成された 3 つの代替 MAC アドレスを指定できます。

- `ldm add-vnet` コマンドを使用します。次の `ldm add-vnet` コマンドは、`domain1` ドメイン上に `vnet1` 仮想ネットワークデバイスを作成し、システムで生成された 3 つの MAC アドレスをそのデバイスで使用できるようにします。

```
primary# ldm add-vnet alt-mac-addr=auto,auto,auto vnet1 primary-vsw0 domain1
```

- `ldm add-vnet` および `ldm set-vnet` コマンドの組み合わせを使用します。次の `ldm add-vnet` および `ldm set-vnet` コマンドは、仮想ネットワークデバイスを作成したあと、既存の仮想ネットワークデバイスにさらに多くの MAC アドレスを割り当てる方法を示しています。

最初のコマンドは、`ldm add-vnet` コマンドを使用して、`domain1` ドメイン上に `vnet1` 仮想ネットワークデバイスを作成します。2 番目のコマンドは、`ldm set-vnet` コマンドを使用して、システムで生成された 3 つの MAC アドレスを `vnet1` 仮想ネットワークデバイスで使用できるようにします。

```
primary# ldm add-vnet vnet1 primary-vsw0 domain1
primary# ldm set-vnet alt-mac-addr=auto,auto,auto vnet1 domain1
```

ドメイン内での Oracle Solaris 11 ゾーンの作成

308 ページの「[仮想ネットワークデバイス上の仮想 NIC の構成](#)」で仮想 NIC を作成したら、使用可能な MAC アドレスに関連付けられたゾーンを作成します。Oracle Solaris ゾーンについての情報は、『[Creating and Using Oracle Solaris Zones](#)』を参照してください。

ゾーンに使用する MAC アドレスを指定するには、`zonecfg` コマンドを使用します。

```
zonecfg:zone-name> set mac-address=[MAC-address,auto]
```

使用可能な MAC アドレスのいずれかを自動的に選択するために値 `auto` を指定するか、または `ldm set-vnet` コマンドで作成した特定の代替 MAC アドレスを指定します。

◆◆◆ 第 13 章

ドメインの移行

この章では、ホストマシンから別のホストマシンにドメインを移行する方法について説明します。
この章では、次の項目について説明します。

- [312 ページの「ドメインの移行の概要」](#)
- [312 ページの「移行処理の概要」](#)
- [313 ページの「ソフトウェアの互換性」](#)
- [313 ページの「移行処理のセキュリティー」](#)
- [315 ページの「ドメイン移行の FIPS 140-2 モード」](#)
- [318 ページの「ドメイン移行の制限」](#)
- [321 ページの「ドメインの移行」](#)
- [322 ページの「アクティブなドメインの移行」](#)
- [330 ページの「バインドされたドメインまたはアクティブでないドメインの移行」](#)
- [321 ページの「予行演習の実行」](#)
- [331 ページの「進行中の移行のモニタリング」](#)
- [332 ページの「進行中の移行の取り消し」](#)
- [332 ページの「移行の失敗からの回復」](#)
- [321 ページの「対話型でない移行の実行」](#)
- [333 ページの「移行の例」](#)

注記 - この章で説明する移行機能を使用するには、最新バージョンの Logical Domains Manager、システムファームウェア、および Oracle Solaris OS を実行している必要があります。以前のバージョンの Oracle VM Server for SPARC を使用する移行については、『[Oracle VM Server for SPARC 3.3 リリースノート](#)』および関連するバージョンの管理ガイドを参照してください。

ドメインの移行の概要

ドメインの移行を使用すると、ホストマシンから別のホストマシンにゲストドメインを移行できます。移行が開始されるマシンのことを、ソースマシンと呼びます。ドメインが移行される先のマシンのことを、ターゲットマシンと呼びます。

移行処理が行われている間に、移行元ドメインが、ソースマシンからターゲットマシンの移行先ドメインに転送されます。

ライブ移行機能を使用すると、パフォーマンスが向上し、アクティブなドメインを実行したまま移行できます。ライブ移行に加えて、バインドされたドメインまたはアクティブでないドメインを移行することもできます。これは、**ゴールド移行**と呼ばれます。

ドメインの移行を使用すると、次のようなタスクを実行できます。

- マシン間での負荷分散
- ゲストドメインの実行を続けながらのハードウェア保守の実行

最適な移行パフォーマンスを実現するには、ソースマシンとターゲットマシンの両方が Logical Domains Manager の最新バージョンを実行していることを確認します。

移行処理の概要

ソースマシン上の Logical Domains Manager はドメインのマイグレーション要求を受け入れ、ターゲットマシン上で実行されている Logical Domains Manager とのセキュアなネットワーク接続を確立します。移行の実行は、この接続が確立されてからです。移行処理は次のフェーズで実行されます。

フェーズ 1: ソースマシンがターゲットマシン内で実行する Logical Domains Manager と接続された後、ソースマシンおよび移行元ドメインについての情報がターゲットマシンに転送されます。この情報を使用して、移行が可能かどうかを判断する一連のチェックが実行されます。実行されるチェックは、移行されるドメインの状態に基づきます。たとえば、移行されるドメインがアクティブになっている場合と、ドメインがバインドされているかアクティブでない場合では、実行される一連のチェックが異なります。

フェーズ 2: フェーズ 1 のすべてのチェックに合格すると、ソースマシンおよびターゲットマシンで移行の準備が行われます。ターゲットマシンで、移行元ドメインを受け入れるドメインが作成さ

れます。移行元ドメインがアクティブでないかバインドされている場合、移行処理はフェーズ 5 に進みます。

フェーズ 3: 移行元ドメインがアクティブの場合、実行時状態の情報がターゲットマシンに転送されます。移行元ドメインは実行を継続し、同時に Logical Domains Manager は OS によってこのドメインに対して行われる変更を追跡します。この情報は、ソースマシン上のハイパーバイザから取得されて、ターゲットマシン上のハイパーバイザにインストールされます。

フェーズ 4: 移行元ドメインが一時停止されます。この時点で、残っているすべての変更された状態情報がターゲットマシンに再コピーされます。このように、ドメインに対する認識可能な中断は、ほとんど、またはまったく発生しません。中断の長さはワークロードによって異なります。

フェーズ 5: ソースマシンの Logical Domains Manager からターゲットマシンの Logical Domains Manager への引き継ぎが行われます。この引き継ぎは、移行先ドメインが実行を再開し (移行元ドメインがアクティブであった場合)、ソースマシンのドメインが破棄されると行われます。この時点以降、移行されたドメインは唯一の動作中のドメインになります。

ソフトウェアの互換性

移行を行うには、次に示すように、ソースマシンとターゲットマシンの両方で互換性のあるソフトウェアが実行されている必要があります。

- 両方のマシンで実行されている Logical Domains Manager のバージョンは、最新バージョンまたは直前にリリースされたバージョンである必要があります。
- ライブ移行をサポートするには、ソースとターゲットの両方のマシンに互換性のあるバージョンのファームウェアがインストールされている必要があります。どちらのマシンも少なくとも、この Oracle VM Server for SPARC ソフトウェアリリースでサポートされる最低限のファームウェアバージョンで実行されている必要があります。

詳細については、[318 ページの「移行におけるバージョン制限」](#)を参照してください。

移行処理のセキュリティー

Oracle VM Server for SPARC には、移行処理用に次のセキュリティー機能があります。

- **認証。** 移行操作は 2 つのマシンで実行されるため、場合によってはユーザーはソースマシンとターゲットマシンの両方で認証されている必要があります。特に、スーパーユーザー以外のユーザーは LDoms Management 権利プロファイルを使用する必要があります。ただし、SSL

証明書を使った移行を実行する場合、ユーザーがターゲットマシンとソースマシンの両方で認証されている必要はなく、別のユーザーを指定することはできません。

`ldm migrate-domain` コマンドを使用すると、オプションで、ターゲットマシン上での認証用の代替ユーザー名を指定できます。この代替ユーザー名を指定しない場合、移行のコマンドを実行するユーザーの名前が使用されます。[例13-3「ゲストドメインの移行と名前の変更」](#)を参照してください。いずれの場合も、ユーザーはターゲットマシンのパスワードを入力するよう求められます。ただし、`-p` オプションによる非対話形式の移行では、パスワード入力が省略されます。[321 ページの「対話型でない移行の実行」](#)を参照してください。

- **暗号化。**Oracle VM Server for SPARC では、機密データが盗まれるのを防ぐため、および追加のハードウェアや専用のネットワークを不要にするために、SSL を使用して移行トラフィックを暗号化します。

暗号化装置があるプラットフォームで、ソースマシンとターゲットマシンの `primary` ドメインに暗号化装置が割り当てられている場合、移行処理が速くなります。この加速が発生するのは、SSL の処理を暗号化装置にオフロードできるためです。

CPU に暗号化命令が搭載されたプラットフォームでは、移行操作の速度が自動的に向上します。この向上が起こるのは、SSL 操作をソフトウェアではなく暗号化命令によって実行できるためです。

- **FIPS 140-2.**ドメイン移行を実行して Oracle Solaris FIPS 140-2 認証 OpenSSL ライブラリを使用するようにシステムを構成できます。[315 ページの「ドメイン移行の FIPS 140-2 モード」](#)を参照してください。

移行のための SSL 証明書の構成

証明書ベースの認証を実行するには、`-c` オプションを `ldm migrate-domain` コマンドで使用します。このオプションは、パスワードファイルおよび代替ユーザーオプションと相互に排他的です。`-c` オプションが指定されていない場合、移行操作はパスワード認証を実行します。

▼ 移行のための SSL 証明書を構成する方法

SSL 証明書を構成するには、このタスクの手順をソースマシンとターゲットマシンの両方の制御ドメイン上で実行する必要があります。

1. `/var/share/ldomsmanager/trust` ディレクトリがまだ存在していない場合は、このディレクトリを作成します。

2. リモート `ldmd` 証明書をローカルの `ldmd` 信頼される証明書ディレクトリにセキュアな状態でコピーします。

リモートの `ldmd` 証明書は、リモートホスト上の `/var/share/ldomsmanager/server.crt` です。ローカルの `ldmd` 信頼される証明書ディレクトリは `/var/share/ldomsmanager/trust` です。リモート証明書ファイル `remote-hostname.pem` を呼び出します。

3. `ldmd` 信頼される証明書ディレクトリから `/etc/certs/CA` へのシンボリックリンクを作成します。
`REMOTE` 変数を `remote-host` に設定します。

```
localhost# ln -s /var/share/ldomsmanager/trust/${REMOTE}.pem /etc/certs/CA/
```

4. `svc:/system/ca-certificates` サービスを再起動します。

```
localhost# svcadm restart svc:/system/ca-certificates
```

5. 構成が使用可能であることを確認します。

```
localhost# openssl verify /var/share/ldomsmanager/trust/${REMOTE}.pem
/var/share/ldomsmanager/trust/remote-hostname.pem: OK
```

6. `ldmd` デーモンを再起動します。

```
localhost# svcadm restart ldmd
```

SSL 証明書の削除

`.pem` ファイルを `/var/share/ldomsmanager/trust` および `/etc/certs/CA` ディレクトリから削除した場合は、まず `svc:/system/ca-certificates` サービス、次に `ldmd` サービスを再起動する必要があります。サービスが再起動されるまで、その `.pem` ファイルを使用する移行は許可されたままとなります。

```
localhost# svcadm restart svc:/system/ca-certificates
localhost# svcadm restart ldmd
```

ドメイン移行の FIPS 140-2 モード

ドメイン移行を実行して Oracle Solaris FIPS 140-2 認証 OpenSSL ライブラリを使用するように Logical Domains Manager を構成できます。Logical Domains Manager が FIPS

140-2 モードの場合、Logical Domains Manager が FIPS 140-2 モードで実行されている別のシステムへのドメインの移行にのみ使用できます。非 FIPS システムへの移行の試行は拒否されます。Logical Domains Manager が FIPS 140-2 モードでない場合は、FIPS 140-2 モードの Logical Domains Manager に移行できません。

Logical Domains Manager を FIPS 140-2 モードで正しく起動するには、FIPS メディエータを有効にする必要があります。手順については、[316 ページの「FIPS 140-2 モードで Logical Domains Manager を実行する方法」](#)を参照してください。

FIPS 140 対応の OpenSSL 実装の詳細およびその使用方法については、『[Managing Encryption and Certificates in Oracle Solaris 11.3](#)』の「[How to Switch to the FIPS 140-Capable OpenSSL Implementation](#)」および『[Oracle SuperCluster M7-8 Owner's Guide: Overview](#)』の「[Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System](#)」を参照してください。

▼ FIPS 140-2 モードで Logical Domains Manager を実行する方法

始める前に Logical Domains Manager を FIPS 140-2 モードで実行する前に、Logical Domains Manager の少なくともバージョン 3.2 が実行されていること、および primary ドメインが少なくとも Oracle Solaris 11.2 OS を実行していることを確認します。

1. FIPS 140-2 OpenSSL メディエータをインストールし、有効にします。

a. 必要に応じて、FIPS 140-2 OpenSSL メディエータをインストールします。

このパッケージは、Oracle Solaris 11.2 OS をインストールするときにデフォルトでインストールされます。

```
# pkg install openssl-fips-140
```

b. 現在の OpenSSL メディエータを一覧表示します。

```
# pkg mediator openssl
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl vendor local default
```

c. 使用可能な OpenSSL メディエータを一覧表示します。

```
# pkg mediator -a openssl
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
```

openssl	vendor	vendor	default
openssl	system	system	fips-140



注意 - 切り替え先の OpenSSL 実装がシステムに存在する必要があります。システムに存在しない実装に切り替えると、システムを使用できなくなる可能性があります。

- d. FIPS 140-2 メディエータを有効にします。

```
# pkg set-mediator -I fips-140 openssl
```

- e. リブートします。

```
# reboot
```

- f. FIPS 140-2 メディエータが設定されていることを確認します。

```
# pkg mediator openssl
MEDIATOR  VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl    system                local      fips-140
```

2. FIPS 140-2 モードを使用するように ldmd デーモンを構成します。

- a. ldmd デーモンを FIPS 140-2 モードにします。

```
# svccfg -s ldms/ldmd setprop ldmd/fips1402_enabled = true
```

- b. ldmd デーモンを再起動します。

```
# svcadm refresh ldmd
# svcadm restart ldmd
```

▼ Logical Domains Manager を FIPS 140-2 モードからデフォルトモードに戻す方法

1. デフォルトの OpenSSL メディエータに戻すことで、FIPS 140-2 OpenSSL メディエータの使用を停止します。

この手順は FIPS 140-2 メディエータがほかのアプリケーションで必要ではない場合のみ実行します。

```
# pkg set-mediator -I default openssl
```

2. リブートします。

```
# reboot
```

3. デフォルトモードを使用するように `ldmd` デーモンを構成します。

```
# svccfg -s ldoms/ldmd setprop ldmd/fips1402_enabled = false
```

4. `ldmd` デーモンを再起動します。

```
# svcadm refresh ldmd  
# svcadm restart ldmd
```

ドメイン移行の制限

次のセクションでは、ドメイン移行の制限について説明します。Logical Domains Manager ソフトウェアおよびシステムファームウェアのバージョンは、移行を許可するために相互に互換している必要があります。また、ドメイン移行を正常に完了するためには、特定の CPU 要件を満たす必要があります。

ソースとターゲットのプラットフォームおよびシステムファームウェアバージョンのすべての組み合わせにおいて、ライブ移行は認定もサポートもされていません。それらのライブ移行を実行できない組み合わせには、代わりにコールド移行を実行できます。

移行におけるバージョン制限

このセクションでは、ライブ移行を実行する際のバージョン制限について説明します。

- **Logical Domains Manager のバージョン。**一方のシステムで最新バージョンの Logical Domains Manager が動作し、他方のシステムで少なくとも直前のバージョンの Logical Domains Manager が動作している場合は、どちらの方向でもライブ移行を実行できます。
- **Oracle Solaris OS のバージョン。**少なくとも Oracle Solaris 10 9/10 OS を実行するゲストドメインのライブ移行を実行できます。Oracle Solaris 10 10/09 OS 以前の Oracle Solaris OS のバージョンを実行するゲストドメインのライブ移行を実行することはできません。これらの古い Oracle Solaris OS バージョンをブートし、このようなドメインのコールド移行を実行することは引き続き可能です。
- **システムファームウェアのバージョン。**一般に、ソースマシンとターゲットマシンの両方が適切な最小バージョンのシステムファームウェアをサポートしている場合は、2 つのシステム間

でのライブ移行を実行できます。[『Oracle VM Server for SPARC 3.3 インストールガイド』の「最小のシステムファームウェアバージョン」](#)を参照してください。

ただし、一部の特定のプラットフォームとファームウェアの組み合わせではライブ移行がサポートされていません。バージョン 8.4 以上、または XCP2210 のシステムファームウェアが実行されているシステムから、それよりも古いバージョンのシステムファームウェアが実行されているシステムへのドメインのライブ移行を試みると、失敗します。この失敗は、新しい方のバージョンと古い方のバージョンのシステムファームウェア間のハイパーバイザ API の不一致のために発生します。この場合は、次のメッセージが表示されます。

```
primary# ldm migrate ldg1 root@target-name
Target Password:
Domain ldg1 is using features of the system firmware that are not supported in
the version of the firmware running on the target machine.
Domain Migration of LDom ldg1 failed
```

ターゲットマシンが SPARC M5-32 システムでないかぎり、バージョン 8.3 のシステムファームウェアが実行されているシステムから、少なくともバージョン 8.4 のシステムファームウェアが実行されているシステムへのドメインのライブ移行を実行できることに注意してください。詳細については、[『Oracle VM Server for SPARC 3.3 リリースノート』の「システムファームウェア 8.3 が実行されている SPARC T4 システムから SPARC T5、SPARC M5、または SPARC M6 システムへのドメインの移行が誤って許可されている」](#)を参照してください。

移行における CPU 間の制限

UltraSPARC T2、UltraSPARC T2 Plus、または SPARC T3 システムと、SPARC T5、SPARC M5、または SPARC M6 システムの間ではライブ移行操作を実行できません。

移行における CPU の制限

移行するドメインで、Oracle Solaris 10 1/13 OS よりも古いバージョンの Oracle Solaris OS が実行されている場合、移行中に次のメッセージが表示されることがあります。

```
Domain domain-name is not running an operating system that is
compatible with the latest migration functionality.
```

Oracle Solaris 10 1/13 OS より前の OS を実行した場合は、次の CPU 要件および制限が適用されます。

- 移行されるドメインにフルコアが割り当てられている必要があります。移行するドメインのスレッドの数がフルコアより少ない場合、移行されたドメインがリブートするまで、追加のスレッドはどのドメインでも使用できません。

- 移行後、移行されたドメインがリブートするまで、移行されたドメインでの CPU の動的再構成 (Dynamic Reconfiguration、DR) は無効になります。リブートした時点で、移行されたドメインで CPU の DR を使用できるようになります。
- ターゲットマシンには、移行されたドメインに必要なスレッドの数に対応できる十分な空きフルコアが存在する必要があります。移行後に、移行されたドメインによってフルコアが部分的にしか使用されない場合、移行されたドメインがリブートするまで、追加のスレッドはどのドメインでも使用できません。

これらの制限は、OpenBoot またはカーネルデバッグで実行されているドメインを移行しようとする際にも適用されます。[OpenBoot PROM からまたはカーネルデバッグで実行中のドメインの移行](#)を参照してください。

perf-counters の設定に関する移行の制限

perf-counters プロパティ値が設定されているドメインの移行を実行するときは注意が必要です。

perf-counters プロパティ値が global に設定されているドメインの移行を実行する前に、perf-counters プロパティ値が global に設定されているほかのドメインがターゲットマシン上にないことを確認します。

移行操作中、perf-counters プロパティは、パフォーマンスアクセス機能がソースマシン、ターゲットマシン、またはその両方で使用可能かどうかに基づき、異なる方法で扱われます。

perf-counters プロパティ値は次のように扱われます。

- ソースマシンのみ。perf-counters プロパティ値はターゲットマシンに伝搬されません。
- ターゲットマシンのみ。移行されるマシンの perf-counters プロパティ値は、perf-counters= と同等になるように更新されます。
- ソースマシンとターゲットマシン。perf-counters プロパティ値は、移行されるドメインからターゲットマシン上の移行されたドメインに伝搬されます。

perf-counters プロパティの詳細については、[376 ページの「パフォーマンスカウンタプロパティの使用」](#)および [ldm\(1M\)](#) のマニュアルページを参照してください。

ドメインの移行

`ldm migrate-domain` コマンドを使用して、あるホストマシンから別のホストマシンへのドメインの移行を開始できます。

注記 - ドメインを移行する場合、`cid` および `mblock` プロパティを使用して割り当てた名前付きリソースは破棄されます。代わりに、ドメインはターゲットシステム上の匿名リソースを使用します。

アクティブなドメインを実行中に移行する方法については、[322 ページの「アクティブなドメインの移行」](#)を参照してください。バインドされたドメインまたはアクティブでないドメインを移行する方法については、[330 ページの「バインドされたドメインまたはアクティブでないドメインの移行」](#)を参照してください。

移行のオプションとオペランドについては、[ldm\(1M\)](#) のマニュアルページを参照してください。

注記 - ドメインの移行が完了したら、新しい構成をソースシステムとターゲットシステムの両方の SP に保存します。その結果、ソースシステムまたはターゲットシステムで電源再投入が実行されれば、移行したドメインの状態は正常です。

予行演習の実行

`ldm migrate-domain` コマンドに `-n` オプションを指定すると、移行のチェックは実行されますが、ドメインの移行は行われません。満たしていない要件がある場合、エラーとして報告されます。予行演習の結果により、実際の移行を試みる前に構成エラーを修正できます。

注記 - 論理ドメインには動的な性質があるため、予行演習が正常に実行されても実際の移行が失敗したり、逆に予行演習が失敗しても実際の移行が成功したりする可能性があります。

対話型でない移行の実行

SSL 証明書方法を使用して、非対話式の移行操作を実行します。非対話式の移行操作を開始するためのレガシーの `ldm migrate-domain -p filename` コマンドの使用は非推奨ですが、このコマンドは引き続き使用可能です。

-p オプションの引数として指定するファイル名には、次の特性が必要です。

- ファイルの最初の行にパスワードが指定されている必要があります。
- パスワードは平文である必要があります。
- パスワードの長さは 256 文字以下である必要があります。

パスワード末尾の改行文字と最初の行のあとのすべての行は無視されます。

ターゲットマシンのパスワードを格納するファイルは、適切にセキュリティー保護する必要があります。この方法でパスワードを格納する場合は、ファイルのアクセス権の設定が 400 または 600 であること、つまり root 所有者 (特権ユーザー) のみがファイルの読み取りまたは書き込みを許可されていることを確認します。

アクティブなドメインの移行

アクティブなドメインを移行するときは、移行元ドメイン、ソースマシン、およびターゲットマシンに対して特定の要件と制限があります。詳細については、[318 ページの「ドメイン移行の制限」](#)を参照してください。

ヒント - ソースマシンとターゲットマシンの両方のプライマリドメインに仮想 CPU をさらに追加することで、移行全体の時間を短縮できます。各 primary ドメインのコア全体を 2 個以上にすることをお勧めしますが、必須ではありません。

移行処理の間にドメインの「時間が遅れ」ます。このタイムロスの問題を軽減するには、移行元ドメインと、Network Time Protocol (NTP) サーバーなどの外部時間ソースを同期します。ドメインを NTP クライアントとして構成すると、ドメインの日時は移行が完了してからすぐに修正されます。

ドメインを Oracle Solaris 10 NTP クライアントとして構成するには、『[System Administration Guide: Network Services](#)』の「[Managing Network Time Protocol \(Tasks\)](#)」を参照してください。ドメインを Oracle Solaris 11 NTP クライアントとして構成する場合は、『[Introduction to Oracle Solaris 11 Network Services](#)』の「[Managing Network Time Protocol \(Tasks\)](#)」を参照してください。

注記 - 移行の最後の中断フェーズで、ゲストドメインがわずかに遅延することがあります。特に、TCP のようにプロトコルに再試行メカニズムが含まれる場合、または NFS over UDP のようにアプリケーションレベルで再試行メカニズムが存在する場合、この遅延によって認識できるようなネットワーク通信の遅延が発生しないようにしてください。ただし、ゲストドメインが Routing Information Protocol (RIP) のようにネットワークの影響を受けやすいアプリケーションを実行する場合、操作を試みるとドメインでわずかな遅延や中断が発生することがあります。この遅延は、ゲストネットワークインタフェースが破損してしまっていて、中断フェーズ中に再作成されるときに短い期間に発生します。

CPU のドメイン移行要件

次に、移行を実行する場合の CPU に対する要件および制限を示します。

- ターゲットマシンには、移行元ドメインによって使用されている仮想 CPU の数に対応できる十分な空き仮想 CPU が存在する必要があります。
- ゲストドメインの `cpu-arch` プロパティにより、プロセッサの種類が異なるシステム間でドメインを移行できます。`cpu-arch` 値を変更するには、ゲストドメインがバウンドまたは非アクティブ状態である必要があります。

サポートされる `cpu-arch` プロパティ値は次のとおりです。

- `native` は CPU 固有のハードウェア機能を使用して、ゲストドメインが同じ CPU タイプのプラットフォーム間でのみ移行できるようにします。`native` はデフォルト値です。
- `migration-class1` は、SPARC T4 以降の SPARC プラットフォーム用の CPU 間移行ファミリです。これらのプラットフォームでは、移行中と移行後のハードウェア暗号化をサポートするため、サポートされる CPU には下限があります。

この値は、UltraSPARC T2、UltraSPARC T2 Plus、または SPARC T3 プラットフォーム、または Fujitsu M10 プラットフォーム とは互換性がありません。

- `sparc64-class1` は、SPARC64 プラットフォーム用の CPU 間移行ファミリです。`sparc64-class1` 値は SPARC64 の命令に基づいているため、`generic` 値より多い命令数が含まれています。そのため、`generic` 値に比べてパフォーマンスに影響しません。

この値は Fujitsu M10 サーバー とのみ互換性があります。

- `generic` はすべてのプラットフォームで使われる最低レベルの一般的な CPU ハードウェア機能を使用して、ゲストドメインが CPU タイプに依存しない移行を実行できるようにします。

次の `isainfo -v` コマンドは、`cpu-arch=generic` のとき、および `cpu-arch=migration-class1` のときに、システムで使用可能な命令を示します。

■ `cpu-arch=generic`

```
# isainfo -v
64-bit sparcv9 applications
    asi_blk_init vis2 vis popc
32-bit sparv applications
    asi_blk_init vis2 vis popc v8plus div32 mul32
```

■ `cpu-arch=migration-class1`

```
# isainfo -v
64-bit sparcv9 applications
    crc32c cbcond pause mont mpmul sha512 sha256 sha1 md5
    camellia des aes ima hpc vis3 fmaf asi_blk_init vis2
    vis popc
32-bit sparv applications
    crc32c cbcond pause mont mpmul sha512 sha256 sha1 md5
    camellia des aes ima hpc vis3 fmaf asi_blk_init vis2
    vis popc v8plus div32 mul32
```

`generic` 値を使用すると、`native` 値の使用と比較して、ゲストドメインのパフォーマンスが低下する場合があります。パフォーマンスの低下は、ゲストドメインが特定の CPU のネイティブハードウェア機能を使用する代わりに、すべてのサポートされている CPU タイプで使用可能な汎用 CPU 機能のみを使用するため、発生する可能性があります。これらの機能を使用しないことで、`generic` 値により、さまざまな機能をサポートする CPU を使用するシステム間で、ドメインのマイグレーションの柔軟性が得られます。

SPARC T4 以上のシステム間でドメインを移行する場合、`cpu-arch=migration-class1` を設定して、ゲストドメインのパフォーマンスを向上させることができます。パフォーマンスは `generic` 値の使用によって向上しますが、`native` 値もゲストドメインの最適なパフォーマンスを実現します。

`cpu-arch` プロパティが `native` に設定されているときに `psrinfo -pv` コマンドを使用すると、次のようにプロセッサタイプを判断できます。

```
# psrinfo -pv
The physical processor has 2 virtual processors (0 1)
```

```
SPARC-T5 (chipid 0, clock 3600 MHz)
```

cpu-arch プロパティーが native 以外の値に設定されている場合、psrinfo -pv の出力にはプラットフォームタイプが表示されません。代わりに、このコマンドは sun4v-cpu CPU モジュールがロードされていることを示します。

```
# psrinfo -pv
```

```
The physical processor has 2 cores and 13 virtual processors (0-12)
```

```
The core has 8 virtual processors (0-7)
```

```
The core has 5 virtual processors (8-12)
```

```
sun4v-cpu (chipid 0, clock 3600 MHz)
```

メモリーの移行要件

ターゲットマシンのメモリー要件は次のとおりです。

- ドメインの移行に対応できる十分な空きメモリー
- 空きメモリーは互換性のあるレイアウトで使用できる必要がある

互換性要件は、SPARC プラットフォームごとに異なります。ただし、少なくとも、サポートされる最大ページサイズを基準にした実アドレスと物理アドレスの配置が、移行されるドメイン内のメモリーブロックごとに保持されている必要があります。

ターゲットマシンでサポートされる最大ページサイズを確認するには、pagesize command を使用します。

少なくとも Oracle Solaris 11.3 OS を実行しているゲストドメインの場合は、移行されるドメインが小さい使用可能な空きメモリーブロックに収まるように、移行されるドメインのメモリーブロックが移行中に自動的に分割される可能性があります。メモリーブロックは、最大ページサイズに整列された境界でのみ分割できます。

オペレーティングシステム、ファームウェア、またはプラットフォームのその他のメモリーレイアウト要件によって、特定の移行中のメモリーブロックの分割が妨げられることがあります。この状況により、ドメインで使用可能な空きメモリーの合計容量が十分な場合でも移行が失敗する可能性があります。

物理 I/O デバイスの移行要件

物理デバイスに直接アクセスするドメインは移行できません。たとえば、I/O ドメインは移行できません。ただし、物理デバイスが関連付けられているデバイスは移行できます。

詳細については、[327 ページの「PCIe エンドポイントデバイスの移行要件」](#)および [328 ページの「PCIe SR-IOV 仮想機能の移行要件」](#)を参照してください。

仮想 I/O デバイスの移行要件

移行元ドメインによって使用されるすべての I/O サービスが、ターゲットマシン上で使用できる必要があります。つまり、次に示す状態になっている必要があります。

- 移行元ドメインで使用されている各仮想ディスクバックエンドが、ターゲットマシンで定義されている必要があります。この共有ストレージは、SAN ディスク (NFS または iSCSI プロトコルを介して使用可能なストレージ) にすることができます。定義する仮想ディスクバックエンドは、ソースマシンと同じボリュームおよびサービス名である必要があります。バックエンドへのパスはソースマシンとターゲットマシンで異なる場合がありますが、同じバックエンドを参照している必要があります。



注意 - ソースマシンとターゲットマシンで仮想ディスクバックエンドへのパスが同じストレージを参照していなくても、移行は成功します。ただし、ターゲットマシンでのドメインの動作は予測不能になり、ドメインを使用できない可能性があります。この問題を解決するには、ドメインを停止し、構成の問題を修正してから、ドメインを再起動します。これらの手順を実行しない場合、ドメインは矛盾した状態のままになる可能性があります。

- 移行元ドメインの各仮想ネットワークデバイスには、対応する仮想ネットワークスイッチがターゲットマシン上に必要です。各仮想ネットワークスイッチの名前は、ソースマシンでデバイスが接続されている仮想ネットワークスイッチと同じである必要があります。

たとえば、移行元ドメインの `vnet0` が `switch-y` という名前の仮想スイッチサービスに接続されている場合、ターゲットマシン上のドメインは、`switch-y` という名前の仮想スイッチサービスを提供する必要があります。

注記 - 移行先ドメインが必要なネットワークリソースにアクセスできるように、ターゲットマシン上の物理ネットワークが正しく構成されている必要があります。正しく設定されていない場合、一部のネットワークサービスが移行完了後のドメインで使用できなくなる可能性があります。

たとえば、ドメインが正しいネットワークサブネットに確実にアクセスできる必要があるような場合です。また、ドメインがターゲットマシンから必要なリモートシステムに到達できるよう、ゲートウェイ、ルーター、またはファイアウォールを適切に構成することが必要な場合もあります。

移行元ドメインによって使用されていた、自動的に割り当てられる範囲内の MAC アドレスは、ターゲットマシンで使用可能である必要があります。

- 仮想コンソール端末集配信装置 (vcc) サービスがターゲットマシン上に存在し、1 つ以上のポートが空いている必要があります。移行時には明示的なコンソール制約は無視されます。移行先ドメイン名をコンソールグループとして使用し、制御ドメインで使用可能な任意の vcc デバイスで使用可能なポートを使用して、移行先ドメインのコンソールが作成されます。制御ドメインで使用可能なポートがない場合は、サービスドメインで使用可能な vcc デバイスで使用可能なポートを使用して、コンソールが作成されます。デフォルトのグループ名との間に衝突がある場合、移行は失敗します。
- 移行されるドメインによって使用されている各仮想 SAN がターゲットマシンで定義されている必要があります。

PCIe エンドポイントデバイスの移行要件

PCIe エンドポイントデバイスが構成されている I/O ドメインでは、ドメインの移行を実行できません。

直接 I/O 機能については、[147 ページの「Creating an I/O Domain by Assigning PCIe Endpoint Devices」](#)を参照してください。

PCIe SR-IOV 仮想機能の移行要件

PCIe SR-IOV 仮想機能が構成されている I/O ドメインでは、ドメインの移行を実行できません。

SR-IOV 機能の詳細については、[第7章「PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成」](#)を参照してください。

NIU ハイブリッド I/O の移行要件

NIU ハイブリッド I/O リソースを使用するドメインを移行できます。NIU ハイブリッド I/O リソースを指定する制約は、ドメインの必須要件ではありません。使用可能な NIU リソースが存在しないマシンにこのようなドメインを移行した場合、制約は維持されますが、この制約が満たされることはありません。

SR-IOV が使われるようになって NIU Hybrid I/O 機能は非推奨になりました。Oracle VM Server for SPARC 3.3 は、この機能を含む最後のソフトウェアリリースです。

暗号化装置の移行要件

暗号化装置のあるプラットフォームで、暗号化装置がバインドされているゲストドメインが暗号化装置の動的再構成 (DR) をサポートしているオペレーティングシステム上で実行されている場合は、そのゲストドメインを移行できます。

移行の開始時に、Logical Domains Manager は移行元ドメインが暗号化装置の DR をサポートしているかどうかを判定します。サポートされている場合、Logical Domains Manager はドメインからすべての暗号化装置の削除を試みます。移行の完了後、移行したドメインに暗号化装置が再度追加されます。

注記 - ターゲットマシンで暗号化装置の制約を満たすことができない場合でも、移行処理はブロックされません。このような場合、移行先ドメインの暗号化装置の数が移行処理前よりも減少する可能性があります。

アクティブなドメインの遅延再構成

ソースマシンまたはターゲットマシン上でアクティブな遅延再構成処理が実行されている場合、移行を開始できません。移行の進行中に、遅延した再構成操作を開始することは許可されません。

アクティブなドメインで電源管理のエラスティックポリシーが有効にされている場合のマイグレーション

ソースマシンとターゲットマシンのいずれかで電源管理 (PM) のエラスティックポリシーが有効になっている場合に、ライブ移行を実行できます。

ほかのドメインの操作

マシンでの移行が完了するまで、移行中のドメインの状態や構成が変更される可能性がある操作はブロックされます。ドメイン自体に対するすべての操作のほか、マシン上のほかのドメインに対するバインドや停止などの操作もブロックされます。

OpenBoot PROM からまたはカーネルデバッグで実行中のドメインの移行

ドメインの移行を実行するには、Logical Domains Manager と、移行元ドメインで実行している Oracle Solaris OS の間の調整が必要です。移行元ドメインが OpenBoot またはカーネルデバッグ (kldb) で実行中の場合、この調整は不可能です。その結果、移行の試みは失敗します。

移行されるドメインが OpenBoot で実行されている場合は、次のメッセージが表示されます。

```
primary# ldm migrate ldg1 system2
Migration is not supported while the domain ldg1 is in the 'OpenBoot Running' state
Domain Migration of LDom ldg1 failed
```

移行されるドメインがカーネルデバッグ (kldb) で実行されている場合は、次のメッセージが表示されます。

```
primary# ldm migrate ldg1 system2
Migration is not supported while the domain ldg1 is in the 'Solaris debugging' state
Domain Migration of LDom ldg1 failed
```

バインドされたドメインまたはアクティブでないドメインの移行

バインドされたドメインまたはアクティブではないドメインは移行時に実行していないので、そのようなドメインに適用されるドメイン移行の制限はわずかです。したがって実行時状態がコピーされないため、SPARC T3 から SPARC T5 プラットフォームや Fujitsu M10 プラットフォーム へというように異なるプラットフォーム間であっても移行できます。

バインドされたドメインを移行するには、ターゲットマシンが移行元ドメインの CPU、メモリ、および入出力の制約を満たす必要があります。このような制約が満たされない場合、移行は失敗します。



注意 - バインドされたドメインを移行する場合、実行時状態情報がターゲットマシンと交換されないため、仮想ディスクバックエンドの `options` および `mpgroup` 値がチェックされません。このチェックは、アクティブなドメインを移行するときに行われます。

アクティブでないドメインの移行には、このような要件はありません。ただし、ターゲットマシンは、あとでバインドが試みられるときに、移行先ドメインの制約を満たす必要があります。満たしていない場合、ドメインのバインドは失敗します。

注記 - ドメインの移行が完了したら、新しい構成をソースシステムとターゲットシステムの両方の SP に保存します。その結果、ソースシステムまたはターゲットシステムで電源再投入が実行されれば、移行したドメインの状態は正常です。

仮想 I/O デバイスの移行要件

アクティブでないドメインの場合、仮想 I/O (VIO) の制約のチェックは実行されません。そのため、VIO サーバーが存在しなくても移行は正常に実行されます。アクティブでないドメインと同様に、そのドメインがバインドされる時点では、VIO サーバーが存在し、使用可能になっている必要があります。

PCIe エンドポイントデバイスの移行要件

PCIe エンドポイントデバイスが構成されている I/O ドメインでは、ドメインの移行を実行できません。この要件は、バインドされたドメインには適用されますが、アクティブでないドメインには適用されません。

直接 I/O (DIO) 機能については、[147 ページの「Creating an I/O Domain by Assigning PCIe Endpoint Devices」](#)を参照してください。

PCIe SR-IOV 仮想機能の移行要件

PCIe SR-IOV 仮想機能が構成されている I/O ドメインでは、ドメインの移行を実行できません。この要件は、バインドされたドメインには適用されますが、アクティブでないドメインには適用されません。

SR-IOV 機能の詳細については、[第7章「PCIe SR-IOV 仮想機能の使用による I/O ドメインの作成」](#)を参照してください。

進行中の移行のモニタリング

移行が行われている間、移行元ドメインと移行先ドメインは、ステータスの出力での表示が異なります。ldm list コマンドの出力には、移行中のドメインの状態が表示されます。

FLAGS フィールドの 6 列目は、次のいずれかの値になります。

- s – 移行元であるドメイン。
- + – 移行先であるドメイン。
- e – ユーザー介入が必要なエラーが発生しました。

次のコマンドは、ldg-src ドメインが移行のソースであることを示しています。

```
# ldm list ldg-src
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg-src   suspended -n---s      1    1G      0.0%  2h 7m
```

次のコマンドは、ldg-tgt ドメインが移行のターゲットであることを示しています。

```
# ldm list ldg-tgt
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg-tgt	bound	-----t	5000	1	1G		

ステータス出力の詳細形式では、移行についての追加情報が示されます。ソースマシンのステータス出力では、処理が完了した割合と、ターゲットマシンおよび移行先ドメインの名前が表示されます。同様に、ターゲットマシンのステータス出力では、処理が完了した割合と、ソースマシンおよび移行元ドメインの名前が表示されます。

次のコマンドでは、ldg-src ドメインの移行処理の進捗状況が表示されます。

```
# ldm list -o status ldg-src
NAME
ldg-src

STATUS
  OPERATION  PROGRESS  TARGET
  migration   17%       system2
```

進行中の移行の取り消し

移行の開始後に、ldm コマンドが KILL 信号で中断された場合、移行処理は中止されます。移行処理が中止された場合、移行先ドメインは破棄され、移行元ドメインはアクティブであった場合は再開されます。ldm コマンドの制御シェルが失われた場合、移行はバックグラウンドで続行されます。

移行処理は、ldm cancel-operation コマンドを使用して、外部から取り消すこともできます。このコマンドによって、進行中の移行が終了され、移行元ドメインはアクティブなドメインとして再開されます。ldm cancel-operation コマンドはソースマシンから開始する必要があります。あるマシン上で実行される移行関連のコマンドは、そのマシンから開始された移行処理に影響を及ぼします。ターゲットマシンは移行処理を制御できません。

移行の失敗からの回復

移行元ドメインがすべての実行時状態情報を移行先ドメインに送信したあと、移行先ドメインがドメインが再開されたことを確認できるまでの間にネットワーク接続が失われた場合、移行操作は終了します。

次の手順を実行して、移行が正常に完了したかどうかを判定する必要があります。

1. 移行先ドメインが処理を正常に再開したかどうかを判定します。移行先ドメインは次の 2 つのいずれかの状態になります。
 - 移行が正常に完了した場合、移行先ドメインは通常の状態になっています。
 - 移行が失敗した場合、ターゲットマシンでは移行先ドメインがクリーンアップされ、削除されます。
2. 移行先ドメインが処理を正常に再開した場合、エラー状態になっているソースマシンのドメインを安全に削除できます。一方、移行先ドメインが存在しない場合は、ソースマシンのドメインはまだマスターバージョンのドメインであり、回復する必要があります。このドメインを回復するには、ソースマシンで `ldm cancel-operation` コマンドを実行します。このコマンドによって、エラー状態がクリアされ、ドメインが元の状態に復元されます。

移行の例

例 13-1 SSL 証明書を使用したゲストドメイン移行の実行

この例は、ldg1 ドメインを system2 という名前のマシンに移行する方法を示しています。移行操作が開始する前に、ソースマシンとターゲットマシンの両方で SSL 証明書を構成済みである必要があります。[314 ページの「移行のための SSL 証明書を構成する方法」](#)を参照してください。

```
# ldm migrate-domain -c ldg1 system2
```

例 13-2 ゲストドメインの移行

この例は、ldg1 ドメインを system2 という名前のマシンに移行する方法を示しています。

```
# ldm migrate-domain ldg1 system2
Target Password:
```

ターゲットマシンのパスワード入力を求められずにこの移行を実行するには、次のコマンドを使用します。

```
# ldm migrate-domain -p pfile ldg1 system2
```

-p オプションには、引数としてファイル名を指定します。指定するファイルには、ターゲットマシンのスーパーユーザーパスワードを指定します。この例では、pfile にターゲットマシン (system2) のパスワードが含まれています。

例 13-3 ゲストドメインの移行と名前の変更

この例では、移行処理の一部としてドメインの名前を変更する方法を示します。移行の一部として、ソースマシン上の `ldg-src` ドメインの名前がターゲットマシン (`system2`) では `ldg-tgt` に変更されます。さらに、ターゲットマシンでの認証には `ldm-admin` ユーザーを使用します。

```
# ldm migrate ldg-src ldm-admin@system2:ldg-tgt
Target Password:
```

例 13-4 移行の失敗メッセージ

この例では、ターゲットマシンが最新の移行機能をサポートしていない場合に表示される可能性があるエラーメッセージを示します。

```
# ldm migrate ldg1 dt212-346
Target Password:
The target machine is running an older version of the domain
manager that does not support the latest migration functionality.

Upgrading to the latest software will remove restrictions on
a migrated domain that are in effect until it is rebooted.
Consult the product documentation for a full description of
these restrictions.

The target machine is running an older version of the domain manager
that is not compatible with the version running on the source machine.

Domain Migration of LDom ldg1 failed
```

例 13-5 ターゲットマシンでのドメインの移行ステータスの取得

この例では、移行の進行中に移行先ドメインでステータスを取得する方法を示します。この例では、ソースマシンは `t5-sys-1` です。

```
# ldm list -o status ldg-tgt
NAME
ldg-tgt

STATUS
  OPERATION    PROGRESS    SOURCE
  migration    55%         t5-sys-1
```

例 13-6 ソースマシンでのドメインの解析可能な移行ステータスの取得

この例では、移行の進行中に移行元ドメインで解析可能なステータスを取得する方法を示します。この例では、ターゲットマシンは `system2` です。

```
# ldm list -o status -p ldg-src
VERSION 1.6
DOMAIN|name=ldg-src|
STATUS
|op=migration|progress=42|error=no|target=system2
```


◆◆◆ 第 14 章 14

リソースの管理

この章では、Oracle VM Server for SPARC システムでリソース管理を実行するための情報を提供します。

この章では、次の項目について説明します。

- [337 ページの「リソースの再構成」](#)
- [339 ページの「リソースの割り当て」](#)
- [340 ページの「CPU の割り当て」](#)
- [344 ページの「ハードパーティションによるシステムの構成」](#)
- [351 ページの「ドメインへの物理リソースの割り当て」](#)
- [356 ページの「メモリーの動的再構成の使用」](#)
- [364 ページの「リソースグループの使用」](#)
- [365 ページの「電源管理の使用」](#)
- [366 ページの「動的なリソース管理の使用」](#)
- [369 ページの「ドメインリソースの一覧表示」](#)
- [376 ページの「パフォーマンスカウンタプロパティの使用」](#)

リソースの再構成

Oracle VM Server for SPARC ソフトウェアを実行しているシステムは、仮想 CPU、仮想 I/O デバイス、暗号化装置、メモリーなどのリソースを構成できます。一部のリソースは実行中のドメインで動的に構成可能ですが、他のリソースは停止中のドメインで構成する必要があります。制御ドメインでリソースを動的に構成できない場合は、まず遅延再構成を開始する必要があります。遅延再構成は、制御ドメインのリブートが完了するまで構成処理を延期します。

動的再構成

動的再構成 (DR)により、オペレーティングシステム (OS) の実行中にリソースの追加や削除を行うことができます。特定のリソースタイプの動的再構成が実行可能かどうかは、論理ドメインで実行されている OS でのサポート状況に依存します。

動的再構成は、次のリソースに対してサポートされています。

- 仮想 CPU – Oracle Solaris 10 OS および Oracle Solaris 11 OS のすべてのバージョンでサポート
- 仮想 I/O デバイス – Oracle Solaris 10 10/08 OS および Oracle Solaris 11 OS 以上でサポート
- 暗号化装置 – Oracle Solaris 10 1/13 OS および Oracle Solaris 11 OS 以上でサポート
- メモリー – [356 ページの「メモリーの動的再構成の使用」](#)を参照
- CPU コア全体 – 『[Oracle VM Server for SPARC 3.3 インストールガイド](#)』の「完全に認定された Oracle Solaris OS バージョン」を参照してください
- 物理 I/O デバイス – サポートなし

DR 機能を使用するには、変更対象のドメインで Logical Domains DR デーモン (drd) が実行されている必要があります。drd(1M) のマニュアルページを参照してください。

遅延再構成

DR 操作がただちに行われるのに対して、遅延再構成操作は次の状況で行われます。

- OS の次回のリブート後
- OS が実行されていない場合、論理ドメインを停止して起動したあと

一般に、遅延再構成操作は制御ドメインに限定されます。ほかのすべてのドメインの場合、リソースの動的再構成が可能でないかぎり、構成を変更するためにドメインを停止する必要があります。

遅延再構成操作は制御ドメインに限定されます。動的に実行できない操作をサポートするために、ルートドメインで遅延再構成が進行中の場合、実行できるコマンドの数が限られます。これらのサブコマンドには、`add-io`、`set-io`、`remove-io`、`create-vf`、および `destroy-vf` があります。また、ルートドメインで `ldm start-reconf` コマンドを実行することもできます。ほかのすべてのドメインの場合、リソースの動的再構成が可能でないかぎり、構成を変更するためにドメインを停止する必要があります。

遅延再構成の進行中は、そのドメインがリブートするまで、または停止して起動するまで、そのドメインに対するその他の再構成リクエストは延期されます。

`ldm cancel-reconf` コマンドは、ドメイン上の遅延再構成操作を取り消します。遅延再構成機能の使用法については、[ldm\(1M\)](#) マニュアルページを参照してください。

注記 - ほかの `ldm remove-*` コマンドで仮想 I/O デバイスの遅延再構成処理をすでに実行している場合は、`ldm cancel-reconf` コマンドを使用できません。`ldm cancel-reconf` コマンドはこの環境で失敗します。

遅延再構成を使用して、制御ドメインでのリソースを減らすことができます。制御ドメインから多数の CPU を削除するには、『[Oracle VM Server for SPARC 3.3 リリースノート](#)』の「[ドメインから多数の CPU を削除すると失敗することがある](#)」を参照してください。制御ドメインから多くのメモリを削除するには、[358 ページ](#)の「[制御ドメインのメモリを減らす](#)」を参照してください。

注記 - `primary` ドメインが遅延再構成状態である場合、Oracle VM Server for SPARC で管理されるリソースは、`primary` ドメインがリブートしたあとでのみ電源管理されます。OS で直接管理されるリソース (Solaris Power Aware Dispatcher で管理される CPU など) は、この状態の影響は受けません。

遅延再構成中に実行を許可される CPU 構成操作は 1 つだけである

遅延再構成中の `primary` ドメイン上で複数の CPU 構成操作を実行しようとしないでください。さらに CPU 構成リクエストを発行しようとすると、それらの要求は拒否されます。

回避方法: 次のいずれかの操作を実行します。

- その遅延再構成を取り消し、別の遅延再構成を開始して、以前の遅延再構成から失われた構成の変更をリクエストします。
- CPU の数が正しくない制御ドメインをリブートし、そのドメインがリブートしたあとで割り当てを修正します。

リソースの割り当て

リソース割り当てメカニズムでは、リソース割り当て制約を使用してバインド時にドメインへリソースを割り当てます。

リソースの割り当ての制約は、ドメインにリソースを割り当てるときに、システムが満たしている必要があるハード要件です。この制約が満たされない場合、リソースの割り当てとドメインのバインドはいずれも失敗します。



注意 - 各ドメインが互いにサービスを提供し合っている 2 つのドメイン間で循環型依存関係を作成しないでください。そのような構成は、一方のドメインが機能停止すると、他方のドメインも使用できなくなるという単一点障害状態を作り出します。また、循環型依存関係の構成では、ドメインが最初にバインドされたあとは、それらをバインド解除できなくなります。

Logical Domains Manager はドメインの循環型依存関係の作成を防ぎません。

循環型依存関係のためにドメインをバインド解除できない場合は、その依存関係の原因となっているデバイスを削除してから、それらのドメインのバインド解除を試みてください。

CPU の割り当て

別々のドメインで同じコアからスレッドを実行した場合、予期しないパフォーマンスの低下が発生する可能性があります。Oracle VM Server for SPARC ソフトウェアは、論理ドメインのバインドプロセス中に、CPU アフィニティー機能を使用して CPU 割り当てを最適化し、これは、ドメインが起動可能になる前に発生します。同じコア内のスレッド間のキャッシュ共有を改善するため、この機能では同じコアからのスレッドが同じ論理ドメインに割り当てられるように試みられます。

ほかに手段がないかぎり、CPU アフィニティーではドメイン間のコア共有が回避されるように試みられます。ドメインにコアの一部が割り当てられ、より多くのストランドがリクエストされる場合は、まずコアの一部からのストランドがバインドされ、次に必要に応じて、リクエストを満たすように別の空きコアが配置されます。

CPU の割り当てメカニズムでは、CPU リソースについて次のような制約が使用されます。

- **コア全体の制約。**この制約は、CPU コアを仮想 CPU ではなくドメインに割り当てるように指定します。ドメインでコアの最大数の制約が有効にされていないかぎり、`ldm set-core` または `ldm set-vcpu` コマンドをそれぞれ使用して、コア全体の制約を追加または削除できます。ドメインは非アクティブか、バインドされているか、アクティブです。ただし、制約を適用する要求を満たす十分なコアが使用可能である必要があります。最悪の状況の例として、別のドメイン要求とコアを共有するドメインがコア全体の制約を要求する場合、要求を満たすために、空きリストからのコアを使用する必要があります。最良の状況の例として、コアの

すべての仮想 CPU がコア境界上にすでに存在するため、CPU リソースを変更せずに、制約が適用されます。

- コアの最大数 (**max-cores**) の制約。この制約は、バインドされているか、またはアクティブなドメインに割り当て可能なコアの最大数を指定します。

▼ コア全体の制約を適用する方法

コアの最大数の制約を設定する前に、ドメインでコア全体の制約が有効になっていることを確認します。

1. ドメインにコア全体の制約を適用します。

```
primary# ldm set-core 1 domain-name
```

2. ドメインでコア全体の制約が有効になっていることを確認します。

```
primary# ldm ls -o resmgt domain-name
```

max-cores が unlimited に設定されていることに注意します。コアの最大数の制約を有効にするまで、ドメインをハードパーティション分割と一緒に使用することはできません。

例 14-1 コア全体の制約の適用

この例は、ldg1 ドメインにコア全体の制約を適用する方法を示しています。最初のコマンドは制約を適用し、2 番目のコマンドはそれが有効にされていることを確認します。

```
primary# ldm set-core 1 ldg1
primary# ldm ls -o resmgt ldg1
NAME
ldg1

CONSTRAINT
  cpu=whole-core
  max-cores=unlimited
```

▼ コアの最大数の制約を適用する方法

コアの最大数の制約を設定する前に、ドメインでコア全体の制約が有効になっていることを確認します。

コアの最大数の制約は、アクティブでないドメインでのみ有効化、変更、または無効化でき、バインドされているドメインまたはアクティブなドメインでは実行できません。制御ドメインでコアの最大数の制約を更新する前に、まず遅延再構成を開始する必要があります。

1. ドメインでコアの最大数の制約を有効にします。

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores domain-name
```

注記 - それらのコアに関連付けられている暗号化装置は、コアの追加に影響されません。このため、システムがドメインに対して自動的に、関連付けられている暗号化装置の追加を行うことはありません。ただし、暗号化装置は、コアの最後の仮想 CPU が削除されるときにのみ、自動的に削除されます。この操作によって、暗号化装置の「孤立」を避けます。

2. コア全体の制約が有効にされていることを確認します。

```
primary# ldm ls -o resmgmt domain-name
```

3. ドメインをバインドし、再起動します。

```
primary# ldm bind domain-name
primary# ldm start domain-name
```

この時点で、ハードパーティション分割を使ったドメインを使用できます。

例 14-2 コアの最大数の制約の適用

この例は、max-cores プロパティを設定してコアの最大数を 3 つのコアに制限し、制約が有効にされていることを確認する方法を示しています。

```
primary# ldm set-domain max-cores=3 ldg1
primary# ldm ls -o resmgmt ldg1
NAME
ldg1

CONSTRAINT
  cpu=whole-core
  max-cores=3
```

この時点で、ハードパーティション分割を使ったドメインを使用できます。

次の例では、バインドされていない非アクティブな ldg1 ドメインからコアの最大数の制約を削除しますが、コア全体の制約はそのまま残します。

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

```
primary# ldm set-domain max-cores=unlimited ldg1
```

また、ldg1 ドメインからコアの最大数の制約とコア全体の制約の両方を削除するには、次のように、コアの代わりに仮想 CPU を割り当てます。

```
primary# ldm set-vcpu 8 ldg1
```

いずれの場合も、ドメインをバインドし、再起動します。

```
primary# ldm bind ldg1
primary# ldm start ldg1
```

コア全体の制約と他のドメイン機能の相互作用

このセクションでは、コア全体の制約と次の機能の相互作用について説明します。

- [343 ページの「CPU の動的再構成」](#)
- [343 ページの「動的なリソース管理」](#)

CPU の動的再構成

コア全体の制約には、CPU の動的再構成 (DR) との完全な互換性があります。ドメインがコア全体の制約で定義されている場合、ldm add-core、ldm set-core、または ldm remove-core コマンドを使用して、アクティブなドメインのコアの数を変更できます。

ただし、バインドされているドメインまたはアクティブなドメインが遅延再構成モードでない場合、コアの数はコアの最大数を超えることはできません。この最大数はコアの最大数の制約により設定され、コア全体の制約が有効化されたときに自動的に有効になります。コアの最大数の制約を満たさない CPU DR 操作はすべて失敗します。

動的なリソース管理

コア全体の制約には、動的リソース管理 (DRM) との完全な互換性があります。

コア全体の制約と DRM の間の期待される相互作用は次のようになります。

- ドメインに対して DRM ポリシーが存在する間は、そのドメインをコア全体の制約付きからコア全体の制約なしに、またはコア全体の制約なしからコア全体の制約付きに切り替えることはできません。たとえば、次のように表示されます。

- ドメインがコア全体の制約付きである場合は、`ldm set-vcpu` コマンドを使用して複数の仮想 CPU を指定したり、コア全体の制約を削除したりすることはできません。
- ドメインがコア全体の制約付きでない場合は、`ldm set-core` コマンドを使用して複数のコア全体を指定したり、コア全体の制約を追加したりすることはできません。
- ドメインがコア全体の制約付きであるときに、`attack`、`decay`、`vcpu-min`、または `vcpu-max` 値を指定する場合、この値はコア全体の倍数である必要があります。

ハードパーティションによるシステムの構成

このセクションでは、Oracle VM Server for SPARC ソフトウェアによるハードパーティション分割およびハードパーティション分割を使用して、Oracle CPU ライセンス要件に準拠する方法を説明します。

ソフトウェアのライセンスに対するオラクルのハードパーティション分割の要件については、[Partitioning: Server/Hardware Partitioning \(http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf\)](http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf)を参照してください。

- **CPU コアと CPU スレッド。**Oracle VM Server for SPARC ソフトウェアは SPARC T シリーズおよび SPARC M シリーズプラットフォーム、および Fujitsu M10 プラットフォームで実行します。これらのシステムで使用されるプロセッサには複数の CPU コアがあり、それぞれに複数の CPU スレッドが含まれます。
- **ハードパーティション分割と CPU 全体のコア。**Oracle VM Server for SPARC 2.0 リリースから、CPU のコア全体の構成を使用して、ハードパーティション分割が適用されます。CPU のコア全体の構成には、個々の CPU スレッドではなく、CPU のコア全体が割り当てられているドメインがあります。デフォルトで、ドメインは CPU スレッドを使用するように構成されています。

ドメインをコア全体の構成でバインドすると、システムは指定された数の CPU コアとそのすべての CPU スレッドをドメインにプロビジョニングします。CPU コア全体の構成を使用すると、バインドされたドメインまたはアクティブなドメインに動的に割り当て可能な CPU コア数が制限されます。
- **Oracle ハードパーティションライセンス。**Oracle ハードパーティションライセンス要件に準拠するには、Oracle VM Server for SPARC 2.0 リリース以上を使用する必要があります。次のように、CPU コア全体を使用する必要もあります。
 - Oracle ハードパーティションライセンスを使用するアプリケーションを実行するドメインは、CPU コア全体で構成する必要があります。

- Oracle ハードパーティションライセンスを使用するアプリケーションを実行しないドメインは、CPU コア全体で構成する必要はありません。たとえば、制御ドメインで Oracle アプリケーションを実行しない場合、そのドメインは CPU のコア全体で構成する必要はありません。

ドメインの構成のチェック

`ldm list-o` コマンドを使用して、ドメインが CPU コア全体で構成されているかどうか、およびドメインに割り当てられている CPU コアを表示する方法を判断します。

- ドメインが CPU コア全体で構成されているかどうかを判断するには

```
primary# ldm list -o resmgt domain-name
```

出力にコア全体の制約が表示され、`max-cores` プロパティでドメインに構成されている CPU コアの最大数を指定していることを確認します。[ldm\(1M\)](#) マニュアルページを参照してください。

次のコマンドは、`ldg1` ドメインが CPU コア全体および最大 5 つのコアで構成されていることを示します。

```
primary# ldm list -o resmgt ldg1
```

```
NAME
```

```
ldg1
```

```
CONSTRAINT
```

```
whole-core
```

```
max-cores=5
```

- ドメインがバインドされると、CPU コアがドメインに割り当てられます。ドメインに割り当てられている CPU コアを表示するには

```
primary# ldm list -o core domain-name
```

次のコマンドは、`ldg1` ドメインに割り当てられているコアを表示します。

```
primary# ldm list -o core ldg1
```

```
NAME
```

```
ldg1
```

```
CORE
CID PCPUSET
1 (8, 9, 10, 11, 12, 13, 14, 15)
2 (16, 17, 18, 19, 20, 21, 22, 23)
```

CPU コア全体によるドメインの構成

このセクションのタスクでは、CPU コア全体で新しいドメインを作成する方法、CPU コア全体で既存のドメインを構成する方法、CPU コア全体で `primary` ドメインを構成する方法について説明します。

CPU コア全体を使用するようにドメインを構成するには、次のコマンドを使用します。

```
ldm set-core number-of-CPU-cores domain
```

このコマンドは、ドメインの CPU コアの最大数、つまり CPU キャップも指定します。[ldm\(1M\)](#) マニュアルページを参照してください。

CPU キャップと CPU コアの割り当ては、個別のコマンドで処理されます。これらのコマンドを使用して、CPU コアを個別に割り当てたり、キャップを設定したり、またはその両方を実行したりすることができます。CPU キャップが設定されていない場合にも、コアに割当て単位を設定できます。ただし、Oracle VM Server for SPARC システムにハードパーティション分割を構成している場合に、システムをこのモードで実行することは許可されません。

- `add-core`、`set-core`、または `rm-core` サブコマンドを使用して、指定した数の CPU コアをドメインに割り当てます。
- `create-domain` または `set-domain` サブコマンドを使用して、CPU キャップを設定し、`max-cores` プロパティ値を指定します。

Oracle VM Server for SPARC システムでハードパーティション分割を構成する場合、キャップを設定する必要があります。

▼ CPU コア全体で新しいドメインを作成する方法

注記 - オプションでコアの最大数の制約を設定する場合にのみ、ドメインを停止し、バインドを解除する必要があります。

1. ドメインを作成します。

```
primary# ldm create domain-name
```

2. ドメインの CPU 全体のコアの数を設定します。

```
primary# ldm set-core number-of-CPU-cores domain
```

3. (オプション) ドメインに `max-cores` プロパティを設定します。

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores domain
```

4. ドメインを構成します。

この構成時には必ず、`ldm add-core`、`ldm set-core`、または `ldm rm-core` コマンドを使用してください。

5. ドメインをバインドし、起動します。

```
primary# ldm bind domain-name
primary# ldm start domain-name
```

例 14-3 2 つの CPU コア全体による新規ドメインの作成

この例では、2 つの CPU コア全体で `ldg1` ドメインを作成します。最初のコマンドでは、`ldg1` ドメインを作成します。2 番目のコマンドは、2 つの CPU コア全体で `ldg1` ドメインを構成します。

この時点で、ドメインの構成をさらに実行することができますが、[346 ページの「CPU コア全体で新しいドメインを作成する方法」](#)の手順 3 で説明する制限を受けます。

3 番目と 4 番目のコマンドは、`ldg1` ドメインのバインドと起動方法を示しており、その時点で、`ldg1` ドメインを使用できます。

```
primary# ldm create ldg1
primary# ldm set-core 2 ldg1
...
primary# ldm bind ldg1
primary# ldm start ldg1
```

▼ CPU コア全体で既存のドメインを構成する方法

ドメインがすでに存在し、CPU スレッドを使用するように構成されている場合は、CPU コア全体を使用するように構成を変更できます。

1. (オプション) ドメインを停止し、バインド解除します。

この段階は、コアの最大数の制約も設定する場合にのみ必要となります。

```
primary# ldm stop domain-name  
primary# ldm unbind domain-name
```

2. **ドメインの CPU 全体のコアの数を設定します。**

```
primary# ldm set-core number-of-CPU-cores domain
```

3. **(オプション) ドメインに max-cores プロパティを設定します。**

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores domain
```

4. **(オプション) ドメインを再バインドし、再起動します。**

この段階は、コアの最大数の制約も設定する場合にのみ必要となります。

```
primary# ldm bind domain-name  
primary# ldm start domain-name
```

例 14-4 4 つの CPU コア全体による既存のドメインの構成

この例では、4 つの CPU コア全体で構成することによって、既存のドメイン `ldg1` の構成を更新します。

```
primary# ldm set-core 4 ldg1
```

▼ CPU コア全体で primary ドメインを構成する方法

CPU スレッドを使用するように primary ドメインが構成されている場合は、CPU コア全体を使用するように構成を変更できます。

1. **(オプション) primary ドメインを遅延再構成モードにします。**

遅延再構成の開始は、`max-cores` プロパティを変更する場合にのみ必要となります。

```
primary# ldm start-reconf primary
```

2. **primary ドメインの CPU コア全体の数を設定します。**

```
primary# ldm set-core number-of-CPU-cores primary
```

3. **(オプション) primary ドメインに max-cores プロパティを設定します。**

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores primary
```

4. (オプション) primary ドメインをリブートします。

システム構成に応じて、適切な手順を使用して、primary ドメインをリブートします。[154 ページの「PCIe エンドポイントを構成した状態のルートドメインのリブート」](#)を参照してください。

ドメインのリブートは、max-cores プロパティを変更する場合にのみ必要となります。

例 14-5 2 つの CPU コア全体を使用した制御ドメインの構成

この例では、primary ドメインに CPU コア全体を構成します。最初のコマンドは、primary ドメインで遅延再構成モードを開始します。2 番目のコマンドは、2 つの CPU コア全体で primary ドメインを構成します。3 つ目のコマンドは max-cores プロパティを 2 に設定し、4 つ目のコマンドは primary ドメインをリブートします。

```
primary# ldm start-reconf primary
primary# ldm set-core 2 primary
primary# ldm set-domain max-cores=2 primary
primary# shutdown -i 5
```

オプションの段階 1 と 4 は、max-cores プロパティを変更する場合にのみ必要となります。

その他の Oracle VM Server for SPARC 機能とハードパーティション化されたシステムとの相互作用

このセクションでは、パーティション化されたシステムがほかの Oracle VM Server for SPARC 機能と通信する方法を説明します。

CPU の動的再構成

CPU コア全体で構成されているドメインで CPU 動的再構成を使用できます。ただし、追加または削除できるのは、個々の CPU スレッドではなく、CPU コア全体のみに限ります。システムのハードパーティション分割状態は CPU 動的再構成機能によって保守します。さらに、CPU コアが動的にドメインに追加された場合、最大数が適用されます。そのため、CPU DR コマンドは CPU の最大数を超過しようとすると、失敗します。

注記 - max-cores プロパティは、ドメインを停止するかバインド解除しないかぎり、変更できません。そのため、コア全体の制約を設定したときに指定した値から、コアの最大数を増やすには、まずドメインを停止し、バインド解除する必要があります。

バインドされているか、アクティブなドメインに CPU コア全体を動的に追加または削除し、バインドされているか、アクティブなドメインの CPU コア全体の数を動的に設定するには、次のコマンドを使用します。

```
ldm add-core number-of-CPU-cores domain
```

```
ldm rm-core number-of-CPU-cores domain
```

```
ldm set-core number-of-CPU-cores domain
```

注記 - ドメインがアクティブでない場合、これらのコマンドはドメインの CPU コアの最大数も調整します。ドメインがバインドされているか、アクティブな場合、これらのコマンドはドメインの CPU コアの最大数に影響しません。

例 14-6 ドメインへの 2 つの CPU コア全体の動的な追加

この例では、2 つの CPU コア全体を ldg1 ドメインに動的に追加する方法を示しています。ldg1 ドメインは CPU コア全体で構成されたアクティブなドメインです。最初のコマンドは ldg1 ドメインがアクティブであることを示しています。2 番目のコマンドは、ldg1 ドメインが CPU コア全体および最大 4 つの CPU コアで構成されていることを示します。3 番目と 5 番目のコマンドは、2 つの CPU コア全体の追加の前と後の、ドメインに割り当てられている CPU コアを示します。4 番目のコマンドは 2 つの CPU コア全体を ldg1 ドメインに動的に追加します。

```
primary# ldm list ldg1
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1      active -n---- 5000  16    2G      0.4%  5d 17h 49m
primary# ldm list -o resmgt ldg1
NAME
ldg1

CONSTRAINT
  whole-core
    max-cores=4
primary# ldm list -o core ldg1
NAME
ldg1

CORE
CID PCPUSET
1 (8, 9, 10, 11, 12, 13, 14, 15)
2 (16, 17, 18, 19, 20, 21, 22, 23)
primary# ldm add-core 2 ldg1
primary# ldm list -o core ldg1
NAME
ldg1
```

```
CORE
CID PCPUSET
1 (8, 9, 10, 11, 12, 13, 14, 15)
2 (16, 17, 18, 19, 20, 21, 22, 23)
3 (24, 25, 26, 27, 28, 29, 30, 31)
4 (32, 33, 34, 35, 36, 37, 38, 39)
```

CPU 動的リソース管理

動的リソース管理 (DRM) を使用して、一部のドメインで CPU リソースを自動的に管理できます。

電源管理

ハードパーティション化されたドメインごとに、個別の電源管理 (PM) ポリシーを設定できます。

ドメインのリブートまたは再バインド

CPU コア全体で構成されているドメインは、ドメインが再起動されるか、システム全体が再起動された場合でも、CPU コア全体で構成されたままになります。ドメインはバインドされている間ずっと、同じ物理 CPU コアを使用します。たとえば、ドメインがリブートされた場合、リブートの前とあとで同じ物理 CPU コアを使用します。または、ドメインがバインドされている間に、システム全体の電源がオフにされた場合、システムの電源が再度オンにされたときに、そのドメインが同じ物理 CPU コアで構成されます。ドメインをバインド解除してから再バインドするか、システム全体を新しい構成で再起動した場合、ドメインは別の物理 CPU コアを使用することがあります。

ドメインへの物理リソースの割り当て

Logical Domains Manager はドメインに割り当てられる物理リソースを自動的に選択します。また、Oracle VM Server for SPARC 3.3 ソフトウェアを使用すれば、上級管理者がドメインに割り当てるか削除する物理リソースを明示的に選択することもできます。

ユーザーが明示的に割り当てるリソースは *名前付きリソース* と呼ばれます。自動的に割り当てられるリソースは *匿名リソース* と呼ばれます。



注意 - 上級管理者でないかぎり、名前付きリソースは割り当てないでください。

物理リソースを制御ドメインおよびゲストドメインに明示的に割り当てることができます。制御ドメインはアクティブなままになるため、物理リソース割り当てを行う前に、オプションで制御ドメインを遅延再構成モードにできます。または、物理割り当てを行うと、遅延再構成が自動的にトリガーされます。[355 ページの「制御ドメインでの物理リソースの管理」](#)を参照してください。物理リソースの制限については、[355 ページの「ドメインでの物理リソース管理の制限」](#)を参照してください。

次の物理リソースを制御ドメインおよびゲストドメインに明示的に割り当てることができます。

■ **物理 CPU。**cid プロパティを設定して、ドメインに物理コア ID を割り当てます。

cid プロパティは、構成するシステムのトポロジについて知識のある管理者のみが使用するようにしてください。この高度な構成機能は、特定の割り当て規則を強制的に適用します。これは、システムのパフォーマンス全体に影響を与える可能性があります。

このプロパティは次のいずれかのコマンドを実行して設定できます。

```
ldm add-core cid=core-ID[,core-ID[,...]] domain-name
```

```
ldm set-core cid=core-ID[,core-ID[,...]] domain-name
```

```
ldm rm-core [-f] cid=core-ID[,core-ID[,...]] domain-name
```

cid プロパティの値としてコア ID を指定した場合、core-ID はドメインに対して明示的に割り当てられるか削除されます。

注記 - ldm add-core コマンドを使用して、(匿名の) コアリソースをすでに使用しているドメインに名前付きのコアリソースを追加することはできません。

■ **物理メモリー。**mblock プロパティを設定して、一連の隣接した物理メモリー領域をドメインに割り当てます。各物理メモリー領域は、物理メモリー開始アドレスとサイズで指定します。

mblock プロパティは、構成するシステムのトポロジについて知識のある管理者のみが使用するようにしてください。この高度な構成機能は、特定の割り当て規則を強制的に適用します。これは、システムのパフォーマンス全体に影響を与える可能性があります。

このプロパティは次のいずれかのコマンドを実行して設定できます。

```
ldm add-mem mblock=PA-start:size[,PA-start:size[,...]] domain-name
```



```
ldm set-mem mblock=PA-start:size[,PA-start:size[,...]] domain-name
```

```
ldm rm-mem mblock=PA-start:size[,PA-start:size[,...]] domain-name
```

ドメインに対して、メモリーブロックを割り当てるか削除するには、mblock プロパティを設定します。有効な値には、コロン (:) で区切られた物理メモリー開始アドレス (PA-start) とメモリーブロックサイズ (size) が含まれます。

注記 - mblock または cid プロパティを設定した場合、動的再構成 (DR) を使用して、実行中のドメイン間でメモリーまたはコアリソースを移動することはできません。ドメイン間でリソースを移動するには、ドメインがバインドされている状態またはアクティブでない状態であることを確認します。制御ドメインの物理リソースの管理については、[355 ページの「制御ドメインでの物理リソースの管理」](#)を参照してください。

注記 - ドメインを移行する場合、cid および mblock プロパティを使用して割り当てた名前付きリソースは破棄されます。代わりに、ドメインはターゲットシステム上の匿名リソースを使用します。

ドメインのリソース制約を表示するには、ldm list-constraints コマンドを使用できます。physical-bindings 制約はドメインに物理的に割り当てられているリソースタイプを指定します。ドメインが作成されると、物理リソースがそのドメインに割り当てられるまで、physical-bindings 制約が解除されます。

physical-bindings 制約は、次の場合に特定の値に設定されます。

- mblock プロパティが指定されている場合 memory
- cid プロパティが指定されている場合 core
- cid と mblock の両方のプロパティが指定されている場合 core,memory

▼ physical-bindings 制約を削除する方法

ゲストドメインの physical-bindings 制約を削除するには、まず物理的にバインドされているリソースをすべて削除する必要があります。

1. ドメインのバインドを解除します。

```
primary# ldm unbind domain-name
```

2. 名前付きリソースを削除します。

- 名前付きコアを削除するには:

```
primary# ldm set-core cid= domain-name
```

- 名前付きメモリーを削除するには:

```
primary# ldm set-mem mblock= domain-name
```

3. CPU またはメモリーリソースを追加します。

- CPU リソースを追加するには:

```
primary# ldm add-vcpu number domain-name
```

- メモリーリソースを追加するには:

```
primary# ldm add-mem size[unit] domain-name
```

4. ドメインを再バインドします。

```
primary# ldm bind domain-name
```

▼ 物理的にバインドされていないすべてのリソースを削除する方法

physical-bindings 制約が設定されていないゲストドメインを制約するには、まず物理的にバインドされていないリソースをすべて削除する必要があります。

1. ドメインのバインドを解除します。

```
primary# ldm unbind domain-name
```

2. リソースの数を 0 に設定します。

- CPU リソースを設定するには:

```
primary# ldm set-core 0 domain-name
```

- メモリーリソースを設定するには:

```
primary# ldm set-mem 0 domain-name
```

3. 物理的にバインドされている CPU またはメモリーリソースを追加します。

- CPU リソースを追加するには:

```
primary# ldm add-core cid=core-ID domain-name
```

- メモリリソースを追加するには:

```
primary# ldm add-mem mblock=PA-start:size domain-name
```

4. ドメインを再バインドします。

```
primary# ldm bind domain-name
```

制御ドメインでの物理リソースの管理

制御ドメインから `physical-bindings` 制約を追加したり削除したりするには、前のセクションの該当する手順に従います。ただし、ドメインのバインドを解除する代わりに、制御ドメインを遅延再構成にします。

匿名リソースと物理的にバインドされている名前付きリソース間の制約を変更すると、自動的に遅延再構成がトリガーされます。`ldm start-reconf primary` コマンドを使用すれば、明示的に遅延再構成になることもできます。

すべての遅延再構成の変更と同様に、プロセスを完了するには、ドメイン (この場合は制御ドメイン) のリブートを実行する必要があります。

注記 - 制御ドメインが遅延再構成モードの場合、制御ドメインで `ldm add-mem` コマンドおよび `ldm rm-mem` コマンドを使用して、無限のメモリ割り当てを実行できます。ただし、`ldm set-core` コマンドを使用して、制御ドメインに対して実行できるコア割り当ては1つだけです。

ドメインでの物理リソース管理の制限

物理リソースの割り当てには、次の制限が適用されます。

- 同じドメイン内で、物理メモリーバインドと非物理メモリーバインド、または物理コアバインドと非物理コアバインドを行うことはできません。
- 同じドメイン内で、非物理メモリーバインドと物理コアバインド、または非物理コアバインドと物理メモリーバインドを使用することはできません。
- ドメインに物理リソースを追加すると、対応するリソースタイプが物理バインドとして制約されます。

- `physical-bindings=core` のドメインに対して匿名 CPU を追加または削除しようとすると、失敗します。
- バインド解除されたリソースの場合、`ldm bind` コマンドを実行した場合にのみ、リソースの割り当てとチェックを行うことができます。
- ドメインから物理メモリーを削除する場合、以前に追加された正確な物理メモリーブロックを削除する必要があります。
- 物理メモリーの範囲はオーバーラップできません。
- 物理リソースをドメインに割り当てるには、`ldm add-core cid=` コマンドまたは `ldm set-core cid=` コマンドのみ使用できます。
- `ldm add-mem mblock=` または `ldm set-mem mblock=` コマンドを使用して複数の物理メモリーブロックを割り当てると、すぐにほかのバインドとの競合が発生していないかどうかについてアドレスとサイズがチェックされます。
- 一部のコアが割り当てられているドメインでは、それらのコアの残りの CPU が空いていて使用可能である場合に、コア全体のセマンティクスを使用できます。

メモリーの動的再構成の使用

メモリーの動的再構成 (DR) は、容量ベースであり、アクティブな論理ドメインでの任意の容量のメモリーの追加または削除を可能にします。

メモリー DR 機能を使用するための要件と制限は次のようになります。

- メモリー DR 操作は任意のドメインで実行できます。ただし、ドメインで同時に実行できるメモリー DR 操作は 1 つだけです。
- メモリー DR 機能では、操作に関与するメモリーのアドレスとサイズが 256M バイト単位であることが要求されます。[359 ページの「メモリー配置」](#)を参照してください。
- 空きメモリープールに存在する、この単位に基づかないメモリーは、メモリー DR 機能を使用してドメインに割り当てることはできません。[360 ページの「単位が調整されていないメモリーの追加」](#)を参照してください。

メモリー DR 操作を使用してドメインのメモリーを再構成できない場合、メモリーを再構成する前にドメインを停止する必要があります。そのドメインが制御ドメインの場合、まず遅延再構成を開始する必要があります。

特定の状況では、Logical Domains Manager は、リクエストされたメモリー割り当てを 8K バイトまたは 4M バイトの倍数のいずれかに切り上げます。次の例は `ldm list-domain -l コ`

マンドの出力例を示しています。ここでは、実際に割り当てられているサイズより制約値の方が小さくなっています。

```
Memory:
Constraints: 1965 M
raddr      paddr5      size
0x10000000 0x291000000 1968M
```

メモリーの追加

ドメインがアクティブの場合、`ldm add-memory` コマンドを使用してドメインにメモリーを動的に追加できます。指定されたメモリーサイズがドメインの現在のメモリーサイズを超えている場合は、`ldm set-memory` コマンドで動的にメモリーを追加することもできます。

メモリーの削除

ドメインがアクティブの場合、`ldm remove-memory` コマンドを使用してドメインから動的にメモリーを削除できます。指定されたメモリーサイズがドメインの現在のメモリーサイズよりも小さい場合、`ldm set-memory` コマンドで動的にメモリーを削除することもできます。

メモリーの削除操作には長い時間が必要な場合があります。指定のドメインに対して `ldm list -l` コマンドを実行すると、`ldm remove-memory` コマンドの進捗状況を追跡できます。

`ldm remove-memory` コマンドを中断する (Control-C を押す) か、`ldm cancel-operation memdr` コマンドを発行して、進行中の削除要求を取り消すことができます。メモリー削除要求を取り消すと、削除要求の未処理の部分、つまりドメインからまだ削除されていない部分のメモリーのみが影響を受けます。

部分的なメモリー DR 要求

動的にメモリーをドメインに追加する、またはドメインから削除するリクエストは、部分的にしか実現されない可能性があります。この結果は、追加または削除にそれぞれ適したメモリーが使用できるかどうかによって異なります。

注記 - ドメインから削除されたメモリーは、他のドメインに追加される前にクリアされます。

制御ドメインのメモリの再構成

メモリ DR 機能を使用して、制御ドメインのメモリを再構成できます。制御ドメインでメモリ DR 要求を実行できない場合、まず遅延再構成を開始する必要があります。

メモリ DR 操作には長い時間を要する場合がありますため、アクティブなドメインから大量のメモリを削除する場合は、メモリ DR 操作は適さない可能性があります。具体的には、システムの初期構成時には、遅延再構成を使用して制御ドメインのメモリを減らすようにしてください。

制御ドメインのメモリを減らす

制御ドメインのメモリを出荷時デフォルト構成から減らすには、メモリ DR ではなく遅延再構成を使用します。このような場合、ホストシステムのメモリすべてを制御ドメインが所有します。アクティブなドメインは、要求されたメモリのすべてを追加できること、またはより一般的には放棄できることを保証されていないため、この目的にメモリ DR 機能は適していません。むしろ、そのドメインで実行されている OS が、要求を最大限に満たすを試みます。さらに、メモリの削除操作には長い時間が必要な場合があります。これらの問題は、制御ドメインのメモリを最初に減らすときのように、大量のメモリ操作が関与する場合にはさらに悪化します。

以上の理由により、次の手順に従って遅延再構成を使用してください。

1. `ldm start-reconf primary` コマンドを使用して、制御ドメインを遅延再構成モードにします。
2. 必要に応じて、制御ドメインにより所有されているホストシステムのリソースを分割します。
3. 必要な場合は、`ldm cancel-reconf` コマンドを使用して手順 2 の操作を元に戻し、やり直します。
4. 制御ドメインをリブートして、再構成の変更を有効にします。

動的再構成と遅延再構成

制御ドメインで遅延再構成が保留されている場合、他のドメインについてのメモリ再構成要求はすべて拒否されます。制御ドメインで遅延再構成が保留中されていない場合、メモリ DR をサポートしないドメインについてのメモリ再構成要求はすべて拒否されます。それらのドメインの場合、リクエストは、遅延再構成リクエストに変換されます。

メモリー配置

メモリー再構成のリクエストは、そのリクエストが適用されるドメインの状態に応じて、配置の要件が異なります。

アクティブなドメインのメモリー配置

- **動的追加および削除。**動的追加と動的削除では、メモリーブロックのアドレスとサイズが 256 MB 単位になります。操作の最小サイズは 256M バイトです。
この単位に基づかない要求や、境界サイズよりも大きい削除要求は拒否されます。
メモリー配置を調整するには、次のコマンドを使用します。
 - `ldm add-memory`。このコマンドで `--auto-adj` オプションを指定した場合、追加されるメモリーの量は 256 MB 単位であるため、ドメインに実際に追加されるメモリーの量が増えることがあります。
 - `ldm remove-memory`。このコマンドで `--auto-adj` オプションを指定した場合、削除されるメモリーの量は 256 MB 単位であるため、ドメインから実際に削除されるメモリーの量が減ることがあります。
 - `ldm set-memory`。このコマンドは、追加または削除操作として扱われます。`--auto-adj` オプションを指定すると、前に説明したように、追加または削除されるメモリーの量は 256M バイト単位になります。この調整により、ドメインのメモリーサイズが増加する可能性があります。
- **遅延再構成。**メモリーブロックのアドレスとサイズは 4M バイト単位です。この単位に基づかない要求を行なった場合、要求は 4M バイト単位に切り上げられます。

バインドされているドメインのメモリー配置

バインドされているドメインの、メモリーブロックのアドレスとサイズは 4M バイト単位です。この単位に基づかない要求を行なった場合、要求は 4M バイト単位に切り上げられます。そのため、ドメインの結果のメモリーサイズが指定よりも大きくなることがあります。

`ldm add-memory`、`ldm set-memory`、および `ldm remove-memory` コマンドに `--auto-adj` オプションを使用すると、結果のメモリーサイズが 256M バイト単位に切り上げられます。そのため、結果のメモリーが指定よりも大きくなる場合があります。

アクティブでないドメインのメモリ配置

ldm add-memory、ldm set-memory、および ldm remove-memory コマンドに --auto-adj オプションを使用すると、結果のメモリサイズが 256M バイト単位に切り上げられます。アクティブでないドメインについては、単位の要件はありません。[359 ページの「バインドされているドメインのメモリ配置」](#)に記載されている制限は、このようなドメインがバインドされたあとで有効になります。

単位が調整されていないメモリの追加

メモリ DR 機能では、アクティブなドメインに動的に追加または削除されるメモリのアドレスとサイズが 256M バイト単位であることが要求されます。このため、この単位に調整されていないメモリを、メモリ DR を使用してアクティブドメインから削除することはできません。

また、空きメモリープールに存在するメモリがこの単位に調整されていない場合、それらをメモリ DR を使用してアクティブなドメインに追加することもできません。

単位が調整されたすべてのメモリが割り当てられたあとで、ldm add-memory コマンドを使用して、単位が調整されていない残りのメモリをバインドされたドメインまたはアクティブでないドメインに追加できます。また、このコマンドを使用して、遅延再構成操作を行い、単位が調整されていない残りのメモリを制御ドメインに追加できます。

次の例は、残り 2 つの 128M バイトのメモリーブロックを primary および ldom1 ドメインに追加する方法を示しています。ldom1 ドメインはバインドされた状態です。

次のコマンドは、制御ドメインで遅延再構成操作を開始します。

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the
primary domain reboots, at which time the new configuration for the
primary domain also takes effect.
```

次のコマンドは、128M バイトのメモリーブロックの 1 つを制御ドメインに追加します。

```
primary# ldm add-memory 128M primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

primary# ldm list
NAME                STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
```



```
primary          active    -ndcv-  SP      8      2688M    0.1%  23d 8h 8m
```

```
primary# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	8	2560M	0.5%	23d 8h 9m
ldom1	bound	-----	5000	1	524M		

次のコマンドは、ほかの 128M バイトのメモリーブロックを ldom1 ドメインに追加します。

```
primary# ldm add-mem 128M ldom1
```

```
primary# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	8	2560M	0.1%	23d 8h 9m
ldom1	bound	-----	5000	1	652M		

メモリー DR の例

次の例は、メモリー DR 操作を実行する方法を示しています。関連する CLI コマンドについては、[ldm\(1M\)](#) のマニュアルページを参照してください。

例 14-7 アクティブなドメインでのメモリー DR 操作

この例は、アクティブなドメイン ldom1 に対し、動的にメモリーの追加と削除を行う方法を示しています。

ldm list の出力の「Memory (メモリー)」フィールドに、各ドメインのメモリーが表示されます。

```
primary# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	0.4%	1d 22h 53m
ldom1	active	-n----	5000	2	2G	0.4%	1d 1h 23m
ldom2	bound	-----	5001	2	200M		

次の ldm add-mem コマンドに指定するメモリーは 256M バイトの倍数である必要があるため、このコマンドはエラーで終了します。次の ldm add-mem コマンドは --auto-adj オプションを使用しているため、追加するメモリーの量として 200M を指定していますが、メモリーの量は 256M バイトに切り上げられます。

```
primary# ldm add-mem 200M ldom1
```

```
The size of memory must be a multiple of 256MB.
```

```
primary# ldm add-mem --auto-adj 200M ldom1
```

```
Adjusting request size to 256M.
```

```
The ldom1 domain has been allocated 56M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  5.0%   8m
ldom1         active -n---- 5000   2    2304M   0.5%   1m
ldom2         bound  ----- 5001   2     200M
```

ldm rm-mem コマンドに指定するメモリーは 256M バイトの倍数である必要があるため、このコマンドはエラーで終了します。同じコマンドに --auto-adj オプションを追加すると、メモリーの量が次の 256M バイトの境界まで切り下げられるため、メモリーの削除は成功します。

```
primary# ldm rm-mem --auto-adj 300M ldom1
Adjusting requested size to 256M.
The ldom1 domain has been allocated 44M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  0.3%   8m
ldom1         active -n---- 5000   2      2G    0.2%   2m
ldom2         bound  ----- 5001   2     200M
```

例 14-8 バインドされたドメインでのメモリー DR 操作

この例は、バインドされたドメイン ldom2 に対してメモリーの追加と削除を行う方法を示しています。

ldm list の出力の「Memory (メモリー)」フィールドに、各ドメインのメモリーが表示されます。最初の ldm add-mem コマンドは、ldom2 ドメインに 100M バイトのメモリーを追加します。次の ldm add-mem コマンドには --auto-adj オプションが指定されているため、さらに 112M バイトのメモリーが ldom2 に動的に追加されます。

ldm rm-mem コマンドは、ldom2 ドメインから 100M バイトを動的に削除します。同じコマンドに --auto-adj オプションを指定して 300M バイトのメモリーを削除すると、メモリーの量は次の 256 M バイトの境界まで切り下げられます。

```
primary# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  0.4%  1d 22h 53m
ldom1         active -n---- 5000   2      2G    0.4%  1d 1h 23m
ldom2         bound  ----- 5001   2     200M
```

```
primary# ldm add-mem 100M ldom2
```

```
primary# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  0.5%  1d 22h 54m
ldom1         active -n---- 5000   2      2G    0.2%  1d 1h 25m
```

```
ldom2          bound      - - - - - 5001    2    300M

primary# ldm add-mem --auto-adj 100M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 112M more memory
than requested because of memory alignment constraints.

primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active     -n-cv-   SP      4     27392M  0.4%  1d 22h 55m
ldom1         active     -n----   5000    2      2G      0.5%  1d 1h 25m
ldom2         bound     - - - - - 5001    2     512M

primary# ldm rm-mem 100M ldom2
primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active     -n-cv-   SP      4     27392M  3.3%  1d 22h 55m
ldom1         active     -n----   5000    2      2G      0.2%  1d 1h 25m
ldom2         bound     - - - - - 5001    2     412M

primary# ldm rm-mem --auto-adj 300M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 144M more memory
than requested because of memory alignment constraints.

primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active     -n-cv-   SP      4     27392M  0.5%  1d 22h 55m
ldom1         active     -n----   5000    2      2G      0.2%  1d 1h 26m
ldom2         bound     - - - - - 5001    2     256M
```

例 14-9 ドメインのメモリーサイズの設定

この例は、ldm set-memory コマンドを使用してドメインに対するメモリーの追加と削除を行う方法を示しています。

ldm list の出力の「Memory (メモリー)」フィールドに、各ドメインのメモリーが表示されます。

```
primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active     -n-cv-   SP      4     27392M  0.5%  1d 22h 55m
ldom1         active     -n----   5000    2      2G      0.2%  1d 1h 26m
ldom2         bound     - - - - - 5001    2     256M
```

次の ldm set-mem コマンドは、primary ドメインのサイズを 3400M バイトに設定する操作を試みます。その結果として発生するエラーは、指定された値が 256M バイト単位でないことを示します。同じコマンドに --auto-adj オプションを追加すると、メモリーの一部を正常に削除し、256M バイトの境界を維持することができます。このコマンドは、ドメインがメモリーを使用しているため、要求されたメモリーのすべてを削除できなかったことを示す警告も発行します。

```
primary# ldm set-mem 3400M primary
An ldm set-mem 3400M command would remove 23992MB, which is not a multiple
of 256MB. Instead, run ldm rm-mem 23808MB to ensure a 256MB alignment.
```

```
primary# ldm set-mem --auto-adj 3400M primary
Adjusting request size to 3.4G.
The primary domain has been allocated 184M more memory
than requested because of memory alignment constraints.
Only 9472M of memory could be removed from the primary domain
because the rest of the memory is in use.
```

次の `ldm set-mem` コマンドは、バインドされた状態にある `ldom2` ドメインのメモリーサイズを 690M バイトに設定します。同じコマンドに `--auto-adj` オプションを追加すると、`ldom2` にさらに 78M バイトのメモリーが動的に追加され、256M バイトの境界が維持されます。

```
primary# ldm set-mem 690M ldom2
primary# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	17920M	0.5%	1d 22h 56m
ldom1	active	-n----	5000	2	2G	0.6%	1d 1h 27m
ldom2	bound	-----	5001	2	690M		

```
primary# ldm set-mem --auto-adj 690M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 78M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	17920M	2.1%	1d 22h 57m
ldom1	active	-n----	5000	2	2G	0.2%	1d 1h 27m
ldom2	bound	-----	5001	2	768M		

リソースグループの使用

リソースグループは、システム内のリソースを表示するための代替方法を提供します。リソースは、プロセッサコア、メモリー、および I/O バス間のベースとなる物理関係に基づいてグループ化されます。異なるプラットフォーム、さらに SPARC T5-2、SPARC T5-8 など、同じサーバーファミリ内の異なるプラットフォーム構成が、ハードウェアの違いを反映する異なるリソースグループを持つことができます。`ldm list-resrc-group` コマンドを使用してリソースグループ情報を表示します。

リソースグループのメンバーシップは、ハードウェア構成によって静的に定義されます。`ldm remove-core` コマンドと `ldm remove-memory` コマンドを使用すると、特定のリソースグループからのリソースを操作できます。

- `remove-core` サブコマンドは、ドメインから削除する CPU コアの数指定します。`-g` オプションを使用してリソースグループを指定した場合、削除対象として選択されるコアはすべて、そのリソースグループのものになります。
- `remove-memory` サブコマンドは、指定された量のメモリーを論理ドメインから削除します。`-g` オプションを使用してリソースグループを指定した場合、削除対象として選択されるメモリーはすべて、そのリソースグループのものになります。

これらのコマンドについては、[ldm\(1M\)](#) のマニュアルページを参照してください。

例については、[375 ページの「リソースグループ情報の一覧表示」](#)を参照してください。

リソースグループの要件と制限

リソースグループ機能は、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでのみ使用できます。

リソースグループ機能には、次の制限があります。

- これは、UltraSPARC T2、UltraSPARC T2 Plus、SPARC T3、および SPARC T4 プラットフォームでは使用できません。
- `ldm list-rsrc-group` コマンドは、これらのサポートされていないプラットフォームに関する情報を何も表示せず、`ldm remove-core` および `ldm remove-memory` コマンドの `-g` バリアン트는機能しません。
- サポートされるプラットフォームでは、*domain-name* の代わりに `_sys_` を指定すると、すべてのシステムメモリーが別のリソースグループ内の空きメモリーに移動されます。このコマンドは、サポートされていないプラットフォームでは無操作です。

電源管理の使用

電源管理 (PM) を有効にするには、まずバージョン 3.0 以上の ILOM ファームウェアで PM ポリシーを設定する必要があります。このセクションでは、Oracle VM Server for SPARC ソフトウェアで PM を使用するために必要な情報をまとめます。

PM 機能および ILOM 機能の詳細は、次を参照してください。

- [第20章「電源管理の使用」](#)

- 『Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI 手順ガイド』の「消費電力のモニタリング」
- 『Oracle Integrated Lights Out Manager (ILOM) 3.0 機能更新およびリリースノート』

動的なリソース管理の使用

ポリシーを使用して、DR 操作を自動的に実行する方法を決定できます。現時点では、仮想 CPU の動的リソース管理を制御するポリシーのみを作成できます。



注意 - CPU の動的なリソース管理 (DRM) には、以下の制限が影響します。

- UltraSPARC T2 および UltraSPARC T2 Plus プラットフォームでは、PM エラスティックポリシーが設定されている場合に、DRM を有効にすることができません。
- UltraSPARC T2 および UltraSPARC T2 Plus プラットフォームでは、DRM が有効になっている間は、パフォーマンスポリシーからエラスティックポリシーへの変更はすべて遅延されます。
- ドメインの移行処理を実行する前に、必ず CPU の DRM を無効にしてください。それ以外の場合、エラーメッセージが表示されます。
- PM エラスティックポリシーが設定されていると、ファームウェアで正規化された利用率がサポートされている場合にのみ DRM を使用できます (8.2.0)。

リソース管理ポリシーでは、論理ドメインで仮想 CPU を自動的に追加および削除できる条件について指定します。ポリシーを管理するには、`ldm add-policy`、`ldm set-policy`、および `ldm remove-policy` コマンドを使用します。

```
ldm add-policy [enable=yes|no] [priority=value] [attack=value] [decay=value]
[elastic-margin=value] [sample-rate=value] [tod-begin=hh:mm[:ss]]
[tod-end=hh:mm[:ss]] [util-lower=percent] [util-upper=percent] [vcpu-min=value]
[vcpu-max=value] name=policy-name domain-name...
ldm set-policy [enable=[yes|no]] [priority=[value]] [attack=[value]] [decay=[value]]
[elastic-margin=[value]] [sample-rate=[value]] [tod-begin=[hh:mm:ss]]
[tod-end=[hh:mm:ss]] [util-lower=[percent]] [util-upper=[percent]] [vcpu-min=[value]]
[vcpu-max=[value]] name=policy-name domain-name...
ldm remove-policy [name=policy-name... domain-name
```

これらのコマンドの詳細およびリソース管理ポリシーの作成については、[ldm\(1M\)](#) マニュアルページを参照してください。

ポリシーは、`tod-begin` プロパティと `tod-end` プロパティで指定された期間の間有効です。`tod-begin` で指定される時間は、`tod-end` で指定される時間より 24 時間制での早い時間である必要があります。デフォルトでは、`tod-begin` および `tod-end` プロパティの値はそれぞれ 00:00:00 と 23:59:59 です。デフォルト値を使用する場合、ポリシーは常に有効です。

ポリシーは、`priority` プロパティの値を使用して、動的リソース管理 (DRM) ポリシーの優先順位を指定します。優先順位の値を使用して、単一ドメイン上の DRM ポリシー間、および単一システム上の DRM が有効なドメイン間の関係が決定されます。数値が低いほど、優先度は高く (良く) なります。有効な値は、1 から 9999 までです。デフォルト値は 99 です。

`priority` プロパティの動作は、次のように、空き CPU リソースのプールの利用度に依存します。

- プール内の空き CPU リソースが使用可能です。この場合、`priority` プロパティによって、1 つのドメインに複数の重複するポリシーが定義されている場合に、どの DRM ポリシーが有効になるのかが決まります。
- プール内の空き CPU リソースが使用可能ではありません。この状況では、`priority` プロパティによって、リソースを同じシステムの低い優先順位のドメインから高い優先順位のドメインに動的に移動できるかどうかを指定します。ドメインの優先順位は、そのドメインで有効になっている DRM ポリシーによって指定された優先順位です。

たとえば、優先順位の高いドメインは、優先順位の低い DRM ポリシーを持つ別のドメインから CPU リソースを取得できます。このリソース取得機能は、DRM ポリシーが有効になっているドメインのみに適用されます。`priority` 値が等しいドメインは、この機能の影響を受けません。そのため、すべてのポリシーに対してデフォルトの優先順位が使用されている場合、ドメインは優先順位の低いドメインからリソースを取得できません。この機能を活用するには、値が等しくならないように `priority` プロパティの値を調整します。

たとえば、`ldg1` ドメインと `ldg2` ドメインの両方で DRM ポリシーが有効になっているとします。`ldg1` ドメインの `priority` プロパティは 1 で、`ldg2` ドメインの `priority` プロパティの値 (2) より優先されます。次の状況では、`ldg1` ドメインは `ldg2` ドメインから CPU リソースを動的に削除して自分に割り当てることができます。

- `ldg1` ドメインにさらに CPU リソースが必要である。
- 空き CPU リソースのプールがすべて使用されている。

このポリシーは、`util-high` および `util-low` プロパティ値を使用して、CPU 使用率の上限と下限を指定します。利用率が `util-high` の値を超えた場合、仮想 CPU の数が `vcpu-min` から `vcpu-max` までの値の範囲に収まるまで、仮想 CPU がドメインに追加されます。利用率が `util-low` の値を下回った場合、仮想 CPU の数が `vcpu-min` から `vcpu-max` までの値の範囲

に収まるまで、仮想 CPU がドメインから削除されます。vcpu-min に達すると、仮想 CPU をそれ以上動的に削除できません。vcpu-max に達すると、仮想 CPU をそれ以上動的に追加できません。

例 14-10 リソース管理ポリシーの追加

たとえば、数週間に渡ってシステムの標準利用率を観測したあと、リソース使用状況を最適化するためにポリシーを設定する場合があります。使用率がもともと高いのは、毎日太平洋標準時の午前 9:00 -午後 6:00、使用率が低いのは、毎日太平洋標準時の午後 6:00 -午前 9:00 です。

このシステム利用率の観測に基づき、システム全体の利用率に従って次の高利用率ポリシーと低利用率ポリシーを作成することにします。

■ **高:** 毎日太平洋標準時の午前 9:00 -午後 6:00

■ **低:** 毎日太平洋標準時の午後 6:00 -午前 9:00

次の `ldm add-policy` コマンドで、高利用率時に `ldom1` ドメインで使用される `high-usage` ポリシーを作成します。

次の `high-usage` ポリシーは次のことを行います。

- `tod-begin` プロパティと `tod-end` プロパティを設定することで、開始時間と終了時間がそれぞれ午前 9:00 と午後 6:00 であることを指定します。
- `util-lower` プロパティと `util-upper` プロパティを設定することで、ポリシー分析を実行する上限と下限がそれぞれ 25 パーセントと 75 パーセントであることを指定します。
- `vcpu-min` プロパティと `vcpu-max` プロパティを設定することで、仮想 CPU の最小数と最大数がそれぞれ 2 と 16 であることを指定します。
- `attack` プロパティを設定することで、任意の 1 回のリソース制御サイクルで追加される仮想 CPU の最大数は 1 であることを指定します。
- `decay` プロパティを設定することで、任意の 1 回のリソース制御サイクルで削除される仮想 CPU の最大数は 1 であることを指定します。
- `priority` プロパティを設定することで、このポリシーの優先順位が 1 であることを指定します。優先順位が 1 であるため、別のポリシーが有効になることが可能であっても、このポリシーが実施されます。
- `name` プロパティを設定することで、ポリシーファイルの名前が `high-usage` であることを指定します。
- `enable` や `sample-rate` など、指定されていないプロパティではデフォルト値を使用します。[ldm\(1M\)](#) マニュアルページを参照してください。


```
primary# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-upper=75 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=high-usage ldom1
```

次の `ldm add-policy` コマンドで、低利用率時に `ldom1` ドメインで使用される `med-usage` ポリシーを作成します。

次の `med-usage` ポリシーは次のことを行います。

- `tod-begin` プロパティと `tod-end` プロパティを設定することで、開始時間と終了時間がそれぞれ午後 6:00 と午前 9:00 であることを指定します。
- `util-lower` プロパティと `util-upper` プロパティを設定することで、ポリシー分析を実行する上限と下限がそれぞれ 10 パーセントと 50 パーセントであることを指定します。
- `vcpu-min` プロパティと `vcpu-max` プロパティを設定することで、仮想 CPU の最小数と最大数がそれぞれ 2 と 16 であることを指定します。
- `attack` プロパティを設定することで、任意の 1 回のリソース制御サイクルで追加される仮想 CPU の最大数は 1 であることを指定します。
- `decay` プロパティを設定することで、任意の 1 回のリソース制御サイクルで削除される仮想 CPU の最大数は 1 であることを指定します。
- `priority` プロパティを設定することで、このポリシーの優先順位が 1 であることを指定します。優先順位が 1 であるため、別のポリシーが有効になることが可能であっても、このポリシーが実施されます。
- `name` プロパティを設定することで、ポリシーファイルの名前が `high-usage` であることを指定します。
- `enable` や `sample-rate` など、指定されていないプロパティではデフォルト値を使用します。[ldm\(1M\)](#) マニュアルページを参照してください。

```
primary# ldm add-policy tod-begin=18:00 tod-end=09:00 util-lower=10 util-upper=50 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=med-usage ldom1
```

ドメインリソースの一覧表示

このセクションでは、`ldm` サブコマンドの構文の使用法、フラグや利用統計情報などの出力項目の定義、および出力として表示されるものと同様の例について説明します。

マシンが読み取り可能な出力

`ldm list` コマンドの出力を使用するスクリプトを作成する場合は、常に `-p` オプションを使用して、マシンが読み取り可能な形式で出力を生成します。

すべての `ldm` サブコマンドの構文の使用法を表示するには、次のコマンドを使用します。

```
primary# ldm --help
```

`ldm` サブコマンドの詳細は、[ldm\(1M\)](#) マニュアルページを参照してください。

フラグの定義

ドメインの出力 (`ldm list`) では、次のフラグを表示できます。コマンドに長形式および解析可能オプション (`-l -p`) を使用すると、`flags=normal,control,vio-service` のように、フラグが省略されずに表示されます。このオプションを使用しない場合は、`-n-cv-` のように略語が表示されます。リストフラグ値は位置に依存します。次の値が、左から順に 6 つの列のそれぞれに表示される可能性があります。

列 1 – ドメインの起動または停止

- `s` – 起動または停止

列 2 – ドメインのステータス

- `n` – 正常
- `t` – 遷移
- `d` – リソースがないために起動できない縮退ドメイン

列 3 – 再構成のステータス

- `d` – 遅延再構成
- `r` – メモリーの動的再構成

列 4 – 制御ドメイン

- `c` – 制御ドメイン

列 5 – サービスドメイン

- `v` – 仮想 I/O サービスドメイン

列 6 – 移行のステータス

- s – 移行のソースドメイン
- t – 移行のターゲットドメイン
- e – 移行時に発生したエラー

利用統計情報の定義

ldm list コマンドの長形式 (-l) オプションによって、仮想 CPU ごとの利用統計情報 (UTIL) が表示されます。この統計情報は、ゲストオペレーティングシステムの代わりに仮想 CPU が実行に費やした時間の割合です。仮想 CPU は、ハイパーバイザに制御が渡される場合を除き、ゲストオペレーティングシステムに代わって実行するものと考えられます。ゲストオペレーティングシステムが仮想 CPU の制御をハイパーバイザに渡さない場合、ゲストオペレーティングシステムの CPU の利用率は常に 100% として表示されます。

論理ドメインについて報告された利用統計情報は、ドメインの仮想 CPU に対する仮想 CPU 利用率の平均です。正規化された利用統計情報 (NORM) は、ゲスト OS の代わりに仮想 CPU が実行に費やした時間の割合です。この値には、サイクルのスキップなどの操作が考慮されます。正規化された仮想化は、システムでバージョン 8.2.0 以上のシステムファームウェアが実行されている場合にのみ使用可能です。

PM でサイクルのスキップ操作が実行されない場合は、100% の利用率が 100% の正規化された利用率と等しくなります。PM でサイクルのスキップを 4/8 に調整すると、100% の利用率が 50% の利用率と等しくなります。これは、CPU の事実上使用可能なサイクル数が半分だけになることを意味します。したがって、完全に利用されている CPU の正規化された利用率は 50% となります。仮想 CPU とゲスト OS の両方の正規化された利用率を表示するには、ldm list または ldm list -l コマンドを使用します。

さまざまなリストの表示

- インストールされている現在のソフトウェアのバージョンを表示するには:

```
primary# ldm -V
```

- すべてのドメインの省略形式のリストを生成するには:

```
primary# ldm list
```

- すべてのドメインの長形式のリストを生成するには:

```
primary# ldm list -l
```

- すべてのドメインの拡張リストを生成するには

```
primary# ldm list -e
```

- すべてのドメインの解析可能でマシンが読み取り可能なリストを生成するには:

```
primary# ldm list -p
```

- 次に示す 1 つ以上の *format* オプションを入力して、出力をリソースのサブセットとして生成できます。1 つ以上の形式を指定する場合、スペースなしでコンマを使用して項目を区切ります。

```
primary# ldm list -o resource[,resource...] domain-name
```

- **console** – 出力には、仮想コンソール (vcons) および仮想コンソール端末集配信装置 (vcc) サービスが含まれます。
- **core** – 出力には、コア全体が割り当てられているドメインについての情報が含まれます。
- **cpu** – 出力には、仮想 CPU (vcpu)、物理 CPU (pcpu)、およびコア ID についての情報が含まれます。
- **crypto** – 暗号化装置の出力には、モジュラー演算ユニット (mau)、およびサポートされているその他の暗号化装置 (Control Word Queue, CWQ) などが含まれます。
- **disk** – 出力には、仮想ディスク (vdisk) および仮想ディスクサーバー (vds) が含まれます。
- **domain-name** – 出力には、変数 (var)、ホスト ID (hostid)、ドメインの状態、フラグ、およびソフトウェアの状態が含まれます。
- **memory** – 出力には、memory が含まれます。
- **network** – 出力には、メディアアクセス制御 (mac) アドレス、仮想ネットワークスイッチ (vsw)、および仮想ネットワーク (vnet) デバイスが含まれます。
- **physio** – 物理入出力には、Peripheral Component Interconnect (pci) およびネットワークインタフェースユニット (niu) が含まれます。
- **resgmt** – 出力には、動的なリソース管理 (DRM) のポリシー情報が含まれ、現在実行中のポリシーと、コア全体の構成に関連する制約の一覧が表示されます。
- **serial** – 出力には、仮想論理ドメインチャネル (vldc) サービス、仮想論理ドメインチャネルクライアント (vldcc)、仮想データプレーンチャネルクライアント (vdpcc)、および仮想データプレーンチャネルサービス (vdpcs) が含まれます。

- **stats** – 出力には、リソース管理ポリシーに関連する統計情報が含まれます。
- **status** – 出力には、進行中のドメインの移行に関連するステータス情報が含まれます。

次の例に、指定可能なさまざまな出力のサブセットを示します。

- 制御ドメインの CPU 情報を一覧表示するには:

```
primary# ldm list -o cpu primary
```

- ゲストドメインのドメイン情報を一覧表示するには:

```
primary# ldm list -o domain ldm2
```

- ゲストドメインのメモリおよびネットワーク情報を一覧表示するには:

```
primary# ldm list -o network,memory ldm1
```

- ゲストドメインの DRM ポリシー情報を一覧表示するには:

```
primary# ldm list -o resgmt,stats ldm1
```

- ドメインの変数とその値を表示するには:

```
primary# ldm list-variable variable-name domain-name
```

たとえば、次のコマンドは、ldg1 ドメインの `boot-device` 変数の値を表示します。

```
primary# ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

- ドメインにバインドされたリソースを一覧表示するには:

```
primary# ldm list-bindings domain-name
```

- SP に格納されている論理ドメイン構成を一覧表示するには:

`ldm list-config` コマンドは、サービスプロセッサに格納されている論理ドメイン構成を一覧表示します。`-r` オプションとともに使用する場合、このコマンドは、制御ドメインに存在する自動保存ファイルの構成を一覧表示します。

構成の詳細については、[381 ページの「ドメイン構成の管理」](#)を参照してください。ほかの例については、[ldm\(1M\)](#) マニュアルページを参照してください。

```
primary# ldm list-config
factory-default
3guests
foo [next poweron]
primary
```

```
reconfig-primary
```

構成名の右にあるラベルの意味は、次のとおりです。

- [current] – 最後にブートされた構成。これは、現在動作している構成に一致する間、つまり再構成を開始するまでの間のみ表示されます。再構成を行なったあとは、注釈が [next poweron] に変更されます。
- [next poweron] – 次回電源を再投入するときに使用される構成。
- [degraded] – 構成は、以前にブートした構成の縮退バージョンです。
- すべてのサーバーリソース (バインドされたリソースおよびバインドされていないリソース) を一覧表示するには:

```
primary# ldm list-devices -a
```

- 割り当て可能なメモリーの量を一覧表示するには:

```
primary# ldm list-devices mem
```

```
MEMORY
```

PA	SIZE
0x14e000000	2848M

- メモリーのどの部分が論理ドメインで利用不可かを判定するには:

```
primary# ldm list-devices -a mem
```

```
MEMORY
```

PA	SIZE	BOUND
0x0	57M	_sys_
0x3900000	32M	_sys_
0x5900000	94M	_sys_
0xb700000	393M	_sys_
0x24000000	192M	_sys_
0x30000000	255G	primary
0x3ff0000000	64M	_sys_
0x3ff4000000	64M	_sys_
0x3ff8000000	128M	_sys_
0x80000000000	2G	ldg1
0x80080000000	2G	ldg2
0x80100000000	2G	ldg3
0x80180000000	2G	ldg4
0x80200000000	103G	

```
0x81bc0000000 145G primary
```

- 使用可能なサービスを一覧表示するには:

```
primary# ldm list-services
```

制約の一覧表示

Logical Domains Manager にとって制約とは、特定ドメインへの割り当てが要求されたリソースを指します。使用可能なリソースに応じて、ドメインに追加するように要求したすべてのリソースを受け取るか、まったく受け取らないかのいずれかです。`list-constraints` サブコマンドは、ドメインに割り当てるように要求したリソースを一覧表示します。

- 1 つのドメインの制約を一覧表示するには:

```
# ldm list-constraints domain-name
```

- 特定のドメインの制約を XML 形式で一覧表示するには:

```
# ldm list-constraints -x domain-name
```

- すべてのドメインの制約を解析可能な形式で一覧表示するには:

```
# ldm list-constraints -p
```

リソースグループ情報の一覧表示

`ldm list-rsrc-group` コマンドを使用すると、リソースグループに関する情報を表示できます。

次のコマンドは、すべてのリソースグループに関する情報を表示します。

```
primary# ldm list-rsrc-group
NAME                CORE MEMORY IO
/SYS/CMU4            12  256G  4
/SYS/CMU5            12  256G  4
/SYS/CMU6            12  128G  4
/SYS/CMU7            12  128G  4
```

ほかの `ldm list-*` コマンドと同様、解析可能な出力、詳細出力、および特定のリソースグループおよびドメインに関する情報を表示するためのオプションを指定できます。詳細については、[ldm\(1M\)](#) のマニュアルページを参照してください。

次の例は、`-l` オプションを使用して、`/SYS/CMU5` リソースグループに関する詳細情報を表示します。

```

primary# ldm list-rsrc-group -l /SYS/CMU5
NAME                                CORE  MEMORY  IO
/SYS/CMU5                          12    256G    4

CORE
  CID                                BOUND
  192, 194, 196, 198, 200, 202, 208, 210  primary
  212, 214, 216, 218                    primary

MEMORY
  PA      SIZE      BOUND
  0xc00000000000  228M    ldg1
  0xc00300000000  127G    primary
  0xc1ffc0000000  64M     _sys_
  0xd00000000000  130816M primary
  0xd1ffc0000000  64M     _sys_

IO
  DEVICE      PSEUDONYM      BOUND
  pci@900     pci_24      primary
  pci@940     pci_25      primary
  pci@980     pci_26      primary
  pci@9c0     pci_27      primary

```

パフォーマンスカウンタプロパティの使用

パフォーマンスレジスタアクセス制御機能により、パフォーマンスレジスタの特定のグループに対するドメインのアクセス権を取得、設定、および設定解除できます。

`ldm add-domain` コマンドと `ldm set-domain` コマンドを使用して、`perf-counters` プロパティの値を指定します。新しい `perf-counters` プロパティ値は、次のリポート時にゲストドメインによって認識されます。`perf-counters` 値が指定されていない場合、この値は `htstrand` になります。[ldm\(1M\)](#) マニュアルページを参照してください。

`perf-counters` プロパティには次の値を指定できます。

<code>global</code>	ドメインに対し、そのドメインに割り当てられたリソースがアクセスできるグローバルパフォーマンスカウンタへのアクセスを許可します。グローバルパフォーマンスカウンタに同時にアクセスできるドメインは 1 つだけです。この値は単独で指定することも、 <code>strand</code> または <code>htstrand</code> 値と一緒に指定することもできます。
<code>strand</code>	ドメインに対し、そのドメインに割り当てられた CPU 上に存在するストランドパフォーマンスカウンタへのアクセスを許可します。この値と <code>htstrand</code> 値と一緒に指定することはできません。

htstrand **strand** 値と同様に動作し、さらにそのドメインに割り当てられた CPU 上のハイパー権限モードイベントの計測を可能にします。この値と **strand** 値を一緒に指定することはできません。

すべてのパフォーマンスカウンタへのすべてのアクセスを無効にするには、**perf-counters=** を指定します。

ハイパーバイザにパフォーマンスアクセス機能がない場合、**perf-counters** プロパティを設定しようとすると失敗します。

ldm list -o domain コマンドと **ldm list -e** コマンドは、**perf-counters** プロパティの値を表示します。パフォーマンスアクセス機能がサポートされていない場合、**perf-counters** 値は出力に表示されません。

例 14-11 ドメインの作成およびそのパフォーマンスレジスタアクセスの指定

global レジスタセットにアクセスできる新しい **ldg0** ドメインを作成します。

```
primary# ldm add-domain perf-counters=global ldg0
```

例 14-12 ドメインのパフォーマンスレジスタアクセスの指定

ldg0 ドメインが **global** および **strand** レジスタセットにアクセスできることを指定します。

```
primary# ldm set-domain perf-counters=global,strand ldg0
```

例 14-13 ドメインがどのレジスタセットにもアクセスできないことの指定

ldg0 ドメインがどのレジスタセットにもアクセスできないことを指定します。

```
primary# ldm set-domain perf-counters= ldg0
```

例 14-14 パフォーマンスアクセス情報の表示

次の例は、**ldm list -o domain** コマンドを使用してパフォーマンスアクセス情報を表示する方法を示します。

- 次の **ldm list -o domain** コマンドは、**global** および **htstrand** パフォーマンス値が **ldg0** ドメインで指定されていることを示します。

```
primary# ldm list -o domain ldg0
NAME      STATE    FLAGS    UTIL
NORM
```

```
ldg0      active      -n----      0.0% 0.0%
```

SOFTSTATE

```
Solaris running
```

UUID

```
062200af-2de2-e05f-b271-f6200fd3eee3
```

HOSTID

```
0x84fb315d
```

CONTROL

```
failure-policy=ignore
```

```
extended-mapin-space=on
```

```
cpu-arch=native
```

```
rc-add-policy=
```

```
shutdown-group=15
```

```
perf-counters=global,htstrand
```

DEPENDENCY

```
master=
```

```
PPRIORITY      4000
```

VARIABLES

```
auto-boot?=false
```

```
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

```
/virtualdevices@100/channel@200/disk@0
```

```
network-boot-arguments=dhcp,hostname=solaris,
```

```
file=http://10.129.241.238:5555/cgi-bin/wanboot-cgi
```

```
pm_boot_policy=disabled=0;ttfc=2000;ttmr=0;
```

- 次の `ldm list -p -o domain` コマンドは、前の例と同じ情報を解析可能な形式で示します。

```
primary# ldm list -p -o domain ldg0
```

```
VERSION 1.12
```

```
DOMAIN|name=ldg0|state=active|flags=normal|util=|norm_util=
```

```
UUID|uuid=4e8749b9-281b-e2b1-d0e2-ef4dc2ce5ce6
HOSTID|hostid=0x84f97452
CONTROL|failure-policy=reset|extended-mapin-space=on|cpu-arch=native|rc-add-policy=|
shutdown-group=15|perf-counters=global,htstrand
DEPENDENCY|master=
VARIABLES
|auto-boot?=false
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
|pm_boot_policy=disabled=0;ttfc=2500000;ttmr=0;
```


ドメイン構成の管理

この章では、ドメイン構成の管理について説明します。

この章では、次の項目について説明します。

- [381 ページの「ドメイン構成の管理」](#)
- [382 ページの「使用可能な構成回復方法」](#)

ドメイン構成の管理

ドメイン「構成」とは、単一のシステム内のすべてのドメインおよびリソース割り当ての詳細を示します。構成は、サービスプロセッサ (SP) に保存および格納し、あとで使用することができます。

SP に構成を保存すると、システムの電源再投入後も構成が保持されます。複数の構成を保存すると、次に電源投入を試みたときにブートする構成を指定できます。

システムに電源を投入すると、SP は選択された構成をブートします。システムは、同じドメインセットを実行し、その構成に指定されている同じ仮想化およびリソース割り当てのパーティション分割を使用します。デフォルトの構成は、最後に保存された構成です。ldm set-spconfig コマンド、または適切な ILOM コマンドを使用して、別の構成を明示的に要求することもできます。



注意 - 常に安定した構成を SP に保存し、XML として保存してください。これらの方法で構成を保存すると、電源の障害後にシステムを回復したり、あとで使用するために保存したりすることができます。[385 ページの「ドメイン構成の保存」](#)を参照してください。

SP に構成を保存するたびに、SP 構成および Logical Domains 制約データベースのローカルコピーが制御ドメインに保存されます。このローカルコピーは、ブートセットと呼ばれます。ブートセットは、システムの電源再投入時に、対応する Logical Domains 制約データベースをロードするために使用されます。

SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでは、制御ドメイン上のブートセットは構成のマスターコピーです。起動時に、Logical Domains Manager は自動的にすべての構成を SP と同期するため、SP 上の構成は常に制御ドメインに格納されたブートセットと同一になります。

注記 - ブートセットには重要なシステムデータが含まれているため、ディスク障害の影響を軽減できるように、制御ドメインのファイルシステムでディスクミラーリングや RAID などのテクノロジーが使用されていることを確認してください。

Physical Domain (物理ドメイン) は、単一 Oracle VM Server for SPARC インスタンスによって管理されるリソースの範囲です。サポートされている SPARC T シリーズプラットフォームのように、物理ドメインは完全な物理システムになることがあります。または、サポートされている SPARC M シリーズプラットフォームのように、システムの全体またはシステムのサブセットになることがあります。

使用可能な構成回復方法

Oracle VM Server for SPARC は、次の構成回復方法をサポートします。

- 自動保存方法。構成が SP で使用できないときに使用されます。
この状況は、次のいずれかの環境で発生する可能性があります。
 - 保存された構成を保持するハードウェアが交換されました
 - 最新の構成の変更を SP に保存することを無視したか、または予期しない電源再投入が発生したため、構成が最新状態ではありません
- `ldm add-domain` 方法。ドメインのサブセットで構成の復元が必要な場合に使用されます
- `ldm init-system` 方法。最終手段としてのみ使用してください。この方法は、SP 上の構成と制御ドメインからの自動保存情報の両方が失われているときのみ使用してください。

自動保存を使用した構成の復元

ドメイン構成が変更された場合は、現在の構成のコピーが制御ドメインに自動的に保存されます。この自動保存処理では、SP に構成が明示的に保存されません。

次の状況でも、自動保存処理はただちに行われます。

- 新しい構成が、SP に明示的に保存されていない場合

■ 構成の変更が、影響を受けるドメインのリブート時まで行われない場合

SP に保存されている構成が失われた場合、この自動保存処理によって構成を回復できます。また、システムの電源再投入後に現在の構成が SP に明示的に保存されなかった場合も、この処理によって構成を回復できます。このような状況では、次のブート用にマークされた構成よりも日付が新しければ、その構成が再起動時に Logical Domains Manager によって復元されます。

注記 - 電源管理、FMA、および ASR イベントでは、自動保存ファイルは更新されません。

自動保存ファイルは、自動または手動で新規または既存の構成に復元できます。デフォルトでは、自動保存構成が、対応する実行中の構成よりも新しい場合、メッセージが Logical Domains ログに書き込まれます。したがって、`ldm add-spconfig -r` コマンドを使用して既存の構成を手動で更新するか、または自動保存データに基づいて新しい構成を作成する必要があります。手動回復を完了するには、このコマンドを使用したあとで電源の再投入を実行する必要があります。

注記 - 遅延再構成が保留中の場合は、構成の変更はただちに自動保存されます。そのため、`ldm list-config -r` コマンドを実行すると、自動保存構成は、現在の構成より新しいものとして表示されます。

`ldm *-spconfig` コマンドを使用して構成を管理する方法と、自動保存ファイルを手動で回復する方法については、[ldm\(1M\)](#) マニュアルページを参照してください。

ブートする構成を選択する方法については、[405 ページの「サービスプロセッサでの Oracle VM Server for SPARC の使用」](#)を参照してください。また、`ldm set-spconfig` コマンドを使用することもできます (このコマンドについては、[ldm\(1M\)](#) のマニュアルページを参照)。

自動回復ポリシー

自動回復ポリシーには、制御ドメインに自動的に保存された 1 つの構成が対応する実行中の構成よりも新しい場合に、構成の回復を処理する方法を指定します。自動回復ポリシーを指定するには、`ldmd` SMF サービスの `autorecovery_policy` プロパティを設定します。このプロパティには、次の値を指定できます。

- `autorecovery_policy=1` - 自動保存構成が、対応する実行中の構成よりも新しい場合に、警告メッセージをログに記録します。これらのメッセージは、`ldmd` SMF ログファイルに記録されます。構成の回復を手動で実行する必要があります。これはデフォルトのポリシーです。

- `autorecovery_policy=2` – 自動保存構成が、対応する実行中の構成よりも新しい場合に、通知メッセージを表示します。この通知メッセージは、毎回の Logical Domains Manager の再起動後に最初に `ldm` コマンドが発行されたときに、いずれかの `ldm` コマンドの出力になります。構成の回復を手動で実行する必要があります。
- `autorecovery_policy=3` – 自動保存構成が、対応する実行中の構成よりも新しい場合に、構成を自動的に更新します。この処理により、次の電源再投入時に使用される SP 構成が書き換えられます。この構成を利用可能にするには、電源再投入をもう 1 回実行する必要があります。この構成は、制御ドメインに保存されている、より新しい構成で更新されます。このアクションは、現在実行中の構成には影響を与えません。この処理は、次の電源再投入時に使用される構成にのみ影響します。新しい構成が SP に保存され、次回システムの電源再投入時にこの構成がブートされるというメッセージも記録されます。これらのメッセージは、`ldmd` SMF ログファイルに記録されます。

▼ 自動回復ポリシーを変更する方法

1. 制御ドメインにログインします。

2. 管理者になります。

Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「[About Using Rights to Control Users and Processes](#)」を参照してください。

3. `autorecovery_policy` プロパティー値を表示します。

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
```

4. `ldmd` サービスを停止します。

```
# svcadm disable ldmd
```

5. `autorecovery_policy` プロパティー値を変更します。

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=value
```

たとえば、自動回復を実行するようにポリシーを設定するには、プロパティー値を 3 に設定します。

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
```

6. `ldmd` サービスをリフレッシュして再起動します。


```
# svcadm refresh ldmd
# svcadm enable ldmd
```

例 15-1 ログへの記録から自動回復への自動回復ポリシーの変更

次の例は、`autorecovery_policy` プロパティの現在の値を表示し、その値を新しい値に変更する方法を示しています。このプロパティの元の値は 1 です。この場合、自動保存の変更はログに記録されます。`ldmd` サービスの停止および再起動には `svcadm` コマンド、プロパティ値の表示および設定には `svccfg` コマンドが使用されます。

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
ldmd/autorecovery_policy integer 1
# svcadm disable ldmd
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
# svcadm refresh ldmd
# svcadm enable ldmd
```

ドメイン構成の保存

1 つのドメインまたはシステム上のすべてのドメインのドメイン構成を保存することができます。

名前付きの物理リソースを除き、次の方法では実際のバインドが保持されません。ただし、その方法では、それらのバインドを作成するために使用した制約が保持されます。構成を保存および復元すると、ドメインは同じ仮想リソースを持ちますが、同じ物理リソースにバインドされるとは限りません。名前付き物理リソースは、管理者によって指定されたとおりにバインドされます。

- 1 つのドメインの構成を保存するには、ドメインの制約を含む XML ファイルを作成します。

```
# ldm list-constraints -x domain-name >domain-name.xml
```

次の例は、`ldg1` ドメインの制約を含む `ldg1.xml` XML ファイルを作成する方法を示しています。

```
# ldm list-constraints -x ldg1 >ldg1.xml
```

- システム上のすべてのドメインの構成を保存するには、すべてのドメインの制約を含む XML ファイルを作成します。

```
# ldm list-constraints -x >file.xml
```

次の例は、システム上のすべてのドメインの制約を含む `config.xml` XML ファイルを作成する方法を示しています。

```
# ldm list-constraints -x >config.xml
```

ドメイン構成の復元

このセクションでは、ゲストドメインおよび制御 (primary) ドメインの XML ファイルからドメイン構成を復元する方法について説明します。

- ゲストドメインのドメイン構成を復元するには、`ldm add-domain -i` コマンドを使用します (このコマンドについては、[386 ページの「XML ファイルからのドメイン構成の復元方法 \(ldm add-domain\)」](#)で説明します)。primary ドメインの制約を XML ファイルに保存することはできますが、そのファイルをこのコマンドの入力として使用することはできません。
- primary ドメインのドメイン構成を復元するには、`ldm init-system` コマンドおよび XML ファイルのリソース制約を使用して、primary ドメインを再構成します。また、`ldm init-system` コマンドを使用して、XML ファイルに記述されているほかのドメインを再構成できます。ただし、それらのドメインは構成が完了しても無効のままである可能性があります。[387 ページの「XML ファイルからのドメイン構成の復元方法 \(ldm init-system\)」](#)を参照してください。

▼ XML ファイルからのドメイン構成の復元方法 (ldm add-domain)

この手順は、ゲストドメインでは有効ですが、制御 (primary) ドメインでは有効ではありません。primary ドメイン、または XML ファイルで記述されているその他のドメインの構成を復元する場合は、[387 ページの「XML ファイルからのドメイン構成の復元方法 \(ldm init-system\)」](#)を参照してください。

1. 入力用に作成した XML ファイルを使用してドメインを作成します。

```
# ldm add-domain -i domain-name.xml
```

2. ドメインをバインドします。

```
# ldm bind-domain [-fq] domain-name
```

-f オプションは、無効なバックエンドデバイスが削除された場合でも、ドメインを強制的にバインドします。-q オプションは、コマンドがより迅速に実行されるように、バックエンドデバイスの検証を無効にします。

3. ドメインを起動します。

```
# ldm start-domain domain-name
```

例 15-2 XML ファイルからの 1 つのドメインの復元

次の例は、1 つのドメインを復元する方法を示しています。最初に、XML ファイルから ldg1 ドメインを復元します。次に、復元した ldg1 ドメインをバインドして再起動します。

```
# ldm add-domain -i ldg1.xml
# ldm bind ldg1
# ldm start ldg1
```

▼ XML ファイルからのドメイン構成の復元方法 (ldm init-system)

この手順では、XML ファイルで ldm init-system コマンドを使用して、以前に保存された構成を再作成する方法について説明します。



注意 - ldm init-system コマンドは、物理 I/O コマンドが使用された構成を正しく復元しない可能性があります。そのようなコマンドは、ldm add-io、ldm set-io、ldm remove-io、ldm create-vf、および ldm destroy-vf です。詳細については、『[Oracle VM Server for SPARC 3.3 リリースノート](#)』の「[ldm init-system コマンドで、物理的な I/O 変更が行われたドメイン構成が正しく復元されないことがある](#)」を参照してください。

始める前に ldm list-constraints -x コマンドを実行して、XML 構成ファイルを作成しておく必要があります。このファイルには、1 つまたは複数のドメインの構成を記載します。

1. **primary** ドメインにログインします。
2. システムが **factory-default** 構成であることを確認します。

```
primary# ldm list-config | grep "factory-default"
factory-default [current]
```

システムが factory-default 構成でない場合は、[429 ページの「出荷時デフォルト構成を復元する方法」](#)を参照してください。

3. **管理者**になります。
Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「[About Using Rights to Control Users and Processes](#)」を参照してください。
4. XML ファイルからドメイン構成を復元します。

```
# ldm init-system [-frs] -i filename.xml
```

構成を有効にするためには、primary ドメインをリブートする必要があります。-r オプションは、構成後に primary ドメインをリブートします。-r オプションを指定しない場合は、手動でリブートを行う必要があります。

-s オプションは、仮想サービス構成 (vds、vcc、および vsw) のみ復元し、リブートしなくても実行できます。

-f オプションは、出荷時のデフォルト構成チェックをスキップし、システム上ですでに構成された内容に関係なく処理を続行します。-f オプションは慎重に使用してください。ldm init-system コマンドは、システムが出荷時のデフォルト構成であると想定するため、XML ファイルで指定された変更が直接適用されます。システムが出荷時のデフォルト以外の構成の場合に -f を使用すると、システムが XML ファイルで指定された構成どおりにならない可能性が高くなります。XML ファイル上の変更と初期構成の組み合わせによっては、1 つ以上の変更がシステムに適用されない可能性があります。

primary ドメインは、ファイルで指定されたとおりに再構成されます。XML ファイル内に構成がある primary 以外の任意のドメインは、再構成されますが、非アクティブのままです。

例 15-3 XML 構成ファイルからのドメインの復元

次の例は、ldm init-system コマンドを使用して factory-default 構成から primary ドメインおよびシステム上のすべてのドメインを復元する方法を示しています。

- primary ドメインを復元します。-r オプションは、構成後に primary ドメインをリブートするために使用します。primary.xml ファイルには、以前に保存した XML ドメイン構成が含まれます。

```
primary# ldm init-system -r -i primary.xml
```

- システム上のすべてのドメインを復元します。システムのドメインを config.xml XML ファイル内の構成で復元します。config.xml ファイルには、以前に保存した XML ドメイン構成が含まれます。ldm init-system コマンドによって、primary ドメインが自動的に再起動されます。ほかのドメインは復元されますが、バインドされずに再起動されます。

```
# ldm init-system -r -i config.xml
```

システムのリブート後、次のコマンドは、ldg1 および ldg2 ドメインをバインドしてリブートします。

```
# ldm bind ldg1
```

```
# ldm start ldg1
# ldm bind ldg2
# ldm start ldg2
```

サービスプロセッサの接続の問題への対処

SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーでは、ldmd サービスと SP の間で通信するために ILOM 相互接続が使用されます。

ドメイン構成を管理するために ldm コマンドを使用しようとし、SP との通信エラーが原因で失敗した場合は、プラットフォームに適用される次の回復手順を実行します。

- **SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバー:** ILOM 相互接続の状態をチェックし、ilomconfig-interconnect サービスを再度有効にします。49 ページの「ILOM 相互接続構成を確認する方法」および 50 ページの「ILOM 相互接続サービスを再度有効にする方法」を参照してください。
- **SPARC T3, SPARC T4, SPARC T5, SPARC M5、および SPARC M6 サーバー:** ldmd サービスを再起動します。

```
primary# svcadm enable ldmd
```

これらの手順で通信の復元に失敗した場合は、SP を再起動します。

◆◆◆ 第 16 章 16

ハードウェアエラーの処理

この章には、Oracle VM Server for SPARC でハードウェアエラーを処理する方法に関する情報が含まれます。

この章では、次の項目について説明します。

- 391 ページの「ハードウェアエラー処理の概要」
- 392 ページの「FMA を使用した障害のあるリソースのブラックリスト登録または構成解除」
- 393 ページの「障害のあるリソースまたは見つからないリソースを検出したあとのドメイン復旧」
- 397 ページの「ドメインの縮退化」
- 397 ページの「I/O リソースを退避としてマーク」

ハードウェアエラー処理の概要

Oracle VM Server for SPARC ソフトウェアでは、SPARC T5 および SPARC M5 で始まる SPARC エンタープライズクラスのプラットフォームに対応した次の RAS 機能が追加されています。

- **障害管理アーキテクチャー (FMA) によるブラックリスト登録。**FMA が障害のある CPU またはメモリーのリソースを検出した場合、Oracle VM Server for SPARC はそれらをブラックリストに登録します。ブラックリストに登録されている障害のあるリソースは、FMA によって修復されたとマークされるまでは、どのドメインにも再割り当てできません。
- **復旧モード。**リソースに障害があるかリソースが見つからないためにブートできないドメイン構成を自動的に復元します。

Fujitsu M10 プラットフォームも回復モードをサポートしています。障害のあるリソースのブラックリスト登録はサポートされていませんが、Fujitsu M10 プラットフォームの自動交換機能には同様の機能が用意されています。

FMA を使用した障害のあるリソースのブラックリスト登録または構成解除

FMA は、障害のあるリソースを検出すると、Logical Domains Manager に通知します。それを受けて、Logical Domains Manager は実行中のドメインのすべてでそのリソースの使用を停止しようと試みます。障害のあるリソースが今後ドメインに割り当てられないようにするために、FMA はブラックリストにリソースを追加します。

Logical Domains Manager は、CPU およびメモリーリソースについてのみブラックリスト登録をサポートしており、I/O リソースについてはサポートしません。

障害のあるリソースが使用中ではない場合、Logical Domains Manager は使用可能なリソースのリストからそのリソースを削除します (`ldm list-devices` の出力で確認できます)。この時点で、以後ドメインに再割り当てできないように、このリソースは内部的に「ブラックリスト登録済み」としてマークされます。

障害のあるリソースが使用中の場合、Logical Domains Manager はリソースを退避しようと試みます。実行中のドメインでサービスの中断を避けるため、Logical Domains Manager は先に CPU またはメモリーの動的再構成を使用して障害のあるリソースの退避を試みます。コアをターゲットとして自由に使える場合、Logical Domains Manager は障害の発生したコアを再マップします。この「ライブ退避」が成功した場合、以後ドメインに割り当てできないように、障害のあるリソースは内部的に「ブラックリスト登録済み」とマークされ、`ldm list-devices` の出力に表示されなくなります。

ライブ退避が失敗した場合、Logical Domains Manager は、障害のあるリソースを内部的に「退避保留」とマークします。このリソースは、影響を受けるゲストドメインがリブートまたは停止されるまで、実行中のドメインで引き続き使用されているため、通常どおりに `ldm list-devices` の出力に表示されます。

影響を受けるゲストドメインが停止またはリブートすると、Logical Domains Manager は障害のあるリソースを退避すること、および以後ドメインに再割り当てできないように、内部的に「ブラックリスト登録済み」とマークすることを試みます。そのようなデバイスは、`ldm` の出力に表示されません。保留中の退避が完了すると、Logical Domains Manager はゲストドメインの起動を試みます。ただし、十分なリソースが使用可能ではないためにゲストドメインが起動できない場合、ゲストドメインは「縮退」とマークされ、手動復旧を実行するためにユーザーの介入を求める次の警告メッセージがログに記録されます。

```
primary# ldm ls
NAME                STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  NORM  UPTIME
```



```
primary      active    -n-cv-  UART   368   2079488M 0.1%  0.0%  16h 57m
gd0          bound    -d- - -  5000    8
```

```
warning: Could not restart domain gd0 after completing pending evacuation.
The domain has been marked degraded and should be examined to see
if manual recovery is possible.
```

システムの電源が再投入されたときに、FMA は障害の残るリソースの退避リクエストを繰り返し、Logical Domains Manager は障害のあるリソースを退避して内部的に「ブラックリスト登録済み」とマークすることによってそれらのリクエストを処理します。

FMA によるブラックリスト登録をサポートするよりも前に、障害のあるリソースが原因でパニック状態になったゲストドメインは、パニックとリブートの無限ループが発生することがあります。ゲストドメインがリブートするときにリソースの退避とブラックリスト登録を使用すると、パニックとリブートのループが回避され、障害のあるリソースの使用が試みられることはなくなります。

障害のあるリソースまたは見つからないリソースを検出したあとのドメイン復旧

SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、または Fujitsu M10 サーバーが電源投入時に障害のあるリソースや欠落しているリソースを検出した場合、Logical Domains Manager は、残りの使用可能なリソースを使用して構成済みドメインを回復しようとします。回復が実行されている間、システム (または SPARC M5、SPARC M6、および SPARC M7 シリーズサーバー上の物理ドメイン) は回復モードにあると呼ばれます。回復モードは、デフォルトで有効になっています。[396 ページの「回復モードの制御」](#)を参照してください。

電源投入時に、次のような状況のために前回選択された電源投入構成がブートできない場合、システムファームウェアは factory-default 構成に戻されます。

- 構成で各 PCIe スイッチ内の I/O トポロジが前回選択された電源投入構成の I/O トポロジと一致しません
- 前回選択された電源投入構成の CPU リソースまたはメモリーリソースがシステムにすでに存在しません。

回復モードが有効になっている場合、Logical Domains Manager は、すべてのアクティブなドメインおよびバインドされたドメインを最後に選択された電源投入構成から回復します。その結果得られる実行中の構成は、縮退構成と呼ばれます。縮退構成は、SP に保存され、新しい SP 構成が保存されるか物理ドメインが電源再投入されるまで、アクティブな構成のままととなります。

注記 - 復旧後、縮退構成はすでに実行中の構成となっているため、縮退構成をアクティブにするために物理ドメインを電源投入する必要はありません。

物理ドメインが電源投入されると、システムファームウェアは最初に前回のオリジナルの電源投入構成をブートしようとします。つまり、見つからないハードウェアや障害のあるハードウェアがその間に交換された場合、システムはオリジナルの通常構成をブートできます。前回選択された電源投入構成がブート可能ではなく、関連する縮退構成が存在する場合、ファームウェアはその縮退構成をブートしようとします。縮退構成がブート可能ではないか存在しない場合、factory-default 構成がブートされ、復旧モードが呼び出されます。

復旧操作は、次の順序で行われます。

- **制御ドメイン。**Logical Domains Manager は、制御ドメインを復旧するために、CPU、メモリ、I/O 構成、および I/O サービスを復元します。

復旧可能なすべてのドメインに必要な CPU またはメモリーの量が使用可能な残りの量よりも大きい場合は、CPU またはコアの数がその他のドメインのサイズに比例して削減されます。たとえば、各ドメインに CPU とメモリーが 25% ずつ割り当てられた 4 ドメインシステムの場合、得られる縮退構成でも CPU とメモリーが各ドメインに 25% ずつ割り当てられます。primary ドメインにもともと最大で 2 つのコア (16 個の仮想 CPU) と 8G バイトのメモリーが搭載されていた場合、制御ドメインのサイズは縮小されません。

その他のドメインに割り当てられているルートコンプレックスおよび PCIe デバイスは、制御ドメインから削除されます。制御ドメインによって所有されているルートコンプレックス上の仮想機能は、再作成されます。制御ドメインに割り当てられていて見つからないルートコンプレックス、PCIe デバイス、物理機能、または仮想機能は、縮退としてマークされます。その後、Logical Domains Manager は制御ドメインをリブートして変更をアクティブにします。

- **ルートドメイン。**制御ドメインがリブートしたら、Logical Domains Manager はルートドメインを復旧します。必要に応じて、CPU およびメモリーの量はその他の復旧可能なドメインに比例して削減されます。ルートコンプレックスは、システムに物理的に存在しなくなった場合に退避としてマークされます。復旧操作中、このルートコンプレックスはドメインに構成されません。ルートドメインは、ルートドメインに割り当てられているルートコンプレックスが少なくとも 1 つ使用可能であれば、復旧されます。使用可能なルートコンプレックスがない場合、ルートドメインは復旧されません。Logical Domains Manager は、ルートドメインをブートし、ルートドメインによって所有されている物理機能上に仮想機能を再作成します。また、ルートドメインによって貸し出されている PCIe スロットを削除します。見つからない PCIe スロット、物理機能、および仮想機能は、退避としてマークされます。ドメインによって提供される仮想 I/O サービスは、可能な場合は再作成されます。

- **I/O ドメイン。**Logical Domains Manager は I/O ドメインを復旧します。システムで見つからない PCIe スロットおよび仮想機能は、退避としてマークされます。必要な I/O デバ

イスがどれも存在しない場合、ドメインは復旧されず、そのドメインの CPU およびメモリーのリソースはその他のドメインが使用できます。ドメインによって提供される仮想 I/O サービスは、可能な場合は再作成されます。

- **ゲストドメイン。**ゲストドメインは、そのドメインを提供するサービスドメインが少なくとも 1 つ復旧された場合のみ、復旧されます。ゲストドメインを復旧できない場合、その CPU およびメモリーのリソースはその他のゲストドメインが使用できます。

可能な場合、元の構成で指定された同量の CPU 数とメモリー量がドメインに割り当てられます。その CPU 数およびメモリー量が使用可能でない場合、それらのリソースは比例して削減され、残りの使用可能なリソースを消費します。

注記 - システムが復旧モードの場合は、`ldm list-*` コマンドのみを実行できます。その他の `ldm` コマンドはすべて、復旧操作が完了するまで無効です。

Logical Domains Manager は、バインドされたドメインおよびアクティブなドメインを復旧しようとします。バインドされていないドメインの既存のリソース構成は、そのままの状態新しい構成にコピーされます。

復旧操作中に使用可能なリソースは、前にブートされた構成よりも少なくなることがあります。その結果、Logical Domains Manager が復旧できるのは、前に構成されたドメインのいくつかのみになることがあります。また、復旧されたドメインには、元の構成のすべてのリソースが含まれないことがあります。たとえば、バインドされたドメインを復旧すると、I/O リソースが前の構成よりも少なくなる可能性があります。ドメインの I/O デバイスが存在しなくなった場合や、親サービスドメインを復旧できなかった場合、そのドメインは復旧されない可能性があります。

復旧モードでは、その手順を Logical Domains Manager SMF ログ `/var/svc/log/ldoms-ldmd:default.log` に記録します。Logical Domains Manager が復旧を開始したとき、制御ドメインをリブートしたとき、復旧が完了したときに、メッセージがシステムコンソールに書き込まれます。



注意 - 復旧されたドメインが完全に動作可能であることは保証されません。ドメインには、OS インスタンスやアプリケーションを実行するために不可欠なリソースが含まれないことがあります。たとえば、復旧されたドメインにはネットワークリソースのみがあり、ディスクリソースがない可能性があります。復旧されたドメインに、アプリケーションを実行するために必要なファイルシステムが見つからない可能性もあります。ドメインでマルチパス化された I/O を使用すると、I/O リソースが見つからない影響が軽減されます。

復旧モードのハードウェアおよびソフトウェア要件

- **ハードウェア要件** – 回復モード機能は、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでサポートされています。
- **ファームウェア要件** – SPARC T5 サーバー、SPARC M5 サーバー、および SPARC M6 サーバーでは、少なくともバージョン 9.1.0.a のシステムファームウェア。SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーでは、少なくともバージョン 9.4.3 のシステムファームウェア。Fujitsu M10 サーバーでは、少なくともバージョン XCP2230 のシステムファームウェア。
- **ソフトウェア要件** – PCIe スロットを貸し出す `primary` 以外のルートドメインは、少なくとも Oracle Solaris 10 1/13 OS または Oracle Solaris 11.2 OS を実行している必要があります。

縮退構成

物理ドメインごとに、SP に縮退構成を 1 つのみ保存することができます。縮退構成がすでに存在する場合、新しく作成される縮退構成と置換されます。

縮退構成を直接対話的操作することはできません。システムファームウェアは、必要に応じて、次の電源投入構成を縮退バージョンで透過的にブートします。このような透過性により、システムは見つからないリソースを再認識した電源再投入後に、元の構成をブートできます。アクティブな構成が縮退構成の場合は、`ldm list-spconfig` の出力で `[degraded]` とマークされます。

アクティブな構成が縮退構成の場合、自動保存機能は無効になります。縮退構成がアクティブなときに新しい構成を SP に保存すると、その新しい構成が通常の非縮退構成になります。

注記 - 前に見つからなかったリソースがその後の電源再投入で再認識された場合、そのリソースは通常構成の内容に影響を及ぼしません。ただし、復旧モードをトリガーした構成をその後選択すると、SP は、すでにすべてのハードウェアが使用可能になっている、元の非縮退構成をブートします。

回復モードの制御

`ldmd/recovery_mode` SMF プロパティは、復旧モードの動作を制御します。回復モードは、デフォルトで有効になっています。

ldmd/recovery_mode プロパティーが存在しないか、または auto に設定されている場合は、回復モードが有効になっています。

ldmd/recovery_mode プロパティーが never に設定されている場合、Logical Domains Manager は何もアクションを実行せずに回復モードを終了し、物理ドメインは出荷時のデフォルト構成を実行します。

注記 - 復旧モードが有効ではないのにシステムファームウェアが復旧モードをリクエストする場合は、リクエストが行われたあとで次のコマンドを発行して復旧モードを有効にします。

```
primary# svccfg -s ldmd setprop ldmd/recovery_mode = astring: auto
primary# svcadm refresh ldmd
primary# svcadm restart ldmd
```

このシナリオでは、システムに対する変更がない場合のみ、つまりシステムが factory-default 構成のままである場合のみ、復旧モードがただちに開始されます。

回復モードを有効にすることに加えて、回復中のルートドメインのブートのタイムアウト値を指定できます。デフォルトでは、ldmd/recovery_mode_boot_timeout プロパティーの値は 30 分です。有効な値は 5 分以上です。

ドメインの縮退化

FMA によるリソースのブラックリスト登録によってリソースが不十分な状態のままであるためにドメインが起動できない場合、そのドメインは縮退としてマークされます。ドメインはバインドされた状態を維持するため、そのドメインに割り当てられている残りのリソースをその他のドメインに再割り当てすることはできません。

I/O リソースを退避としてマーク

復旧モードで見つからないことが検出された I/O リソースは、退避としてマークされ、ldm リスト出力ではアスタリスク (*) が表示されます。

◆◆◆ 第 17 章

その他の管理タスクの実行

この章では、ここまでの章では説明していない Oracle VM Server for SPARC ソフトウェアの使用とタスクに関する情報について説明します。

この章では、次の項目について説明します。

- 400 ページの「CLI での名前の入力」
- 400 ページの「`/etc/system` ファイルのプロパティ値の更新」
- 401 ページの「ネットワークを介したゲストコンソールへの接続」
- 401 ページの「コンソールグループの使用」
- 402 ページの「負荷が大きいドメインの停止処理がタイムアウトする可能性」
- 403 ページの「Oracle VM Server for SPARC による Oracle Solaris OS の運用」
- 405 ページの「サービスプロセッサでの Oracle VM Server for SPARC の使用」
- 406 ページの「ドメインの依存関係の構成」
- 410 ページの「CPU およびメモリアドレスのマッピングによるエラー発生箇所の確認」
- 413 ページの「ユニバーサル固有識別子の使用」
- 413 ページの「仮想ドメイン情報コマンドと API」
- 414 ページの「論理ドメインチャネルの使用」
- 418 ページの「多数のドメインのブート」
- 419 ページの「Oracle VM Server for SPARC システムの正常な停止と電源の再投入」
- 420 ページの「Logical Domains 変数の永続性」
- 421 ページの「割り込み制限の調整」
- 424 ページの「ドメイン I/O の依存関係の一覧表示」
- 425 ページの「Logical Domains Manager デーモンの有効化」
- 426 ページの「自動保存構成データの保存および復元」
- 428 ページの「出荷時デフォルト構成とドメインの無効化」

CLI での名前の入力

次のセクションでは、Logical Domains Manager CLI で名前を入力する場合の制限について説明します。

- ファイル名 (*file*) と変数名 (*var-name*)
 - 最初の文字は、英字、数字、またはスラッシュ (/) である必要があります。
 - 以降の文字は、英字、数字、または句読点である必要があります。
- 仮想ディスクサーバー *backend* および仮想スイッチデバイス名
名前は、英字、数字、または句読点を含む必要があります。
- 構成名 (*config-name*)
サービスプロセッサ (SP) に格納されている構成に割り当てる論理ドメイン構成名 (*config-name*) は、64 文字以下である必要があります。
- その他のすべての名前
論理ドメイン名 (*domain-name*)、サービス名 (*vswitch-name*、*service-name*、*vdpcs-service-name*、および *vcc-name*)、仮想ネットワーク名 (*if-name*)、仮想ディスク名 (*disk-name*) など、その他の名前は、次の形式である必要があります。
 - 最初の文字は、英字または数字である必要があります。
 - 以降の文字は、英字、数字、または次のいずれかの文字 -_+#. : ; ~ () である必要があります。

/etc/system ファイルのプロパティ値の更新

/etc/system ファイルを手動で変更しないでください。このファイルは、/etc/system.d ディレクトリ内のファイルでチューニングプロパティ値が指定されている場合、リブート時に自動的に生成されます。

▼ チューニングプロパティ値を追加または変更する方法

1. 既存の /etc/system ファイルおよび /etc/system.d のファイルでチューニングプロパティ値を検索します。
たとえば、`vds:vd_volume_force_slice` プロパティの値を指定するには、プロパティがすでに設定されているかどうかを確認します。


```
# grep 'vds:vd_volume_force_slice' /etc/system /etc/system.d/*
```

2. プロパティー値を更新します。

- プロパティーが `/etc/system.d` のファイルのいずれかで見つかった場合は、既存のファイルでプロパティー値を更新します。
- プロパティーが `/etc/system` ファイルで見つかった、またはプロパティーが見つからない場合は、次の例のような名前で `/etc/system.d` ディレクトリ内にファイルを作成します。

```
/etc/system.d/com.company-name:ldoms-config
```

ネットワークを介したゲストコンソールへの接続

`vntsd(1M)` の SMF マニフェストで `listen_addr` プロパティーが制御ドメインの IP アドレスに設定されている場合は、ネットワークを介してゲストコンソールに接続できます。たとえば、次のように表示されます。

```
$ telnet hostname 5001
```

注記 - コンソールへのネットワークアクセスを有効にすることには、セキュリティ上の問題があります。すべてのユーザーがコンソールに接続できるようになるため、デフォルトではこの設定は無効になっています。

サービス管理機能マニフェストは、サービスが記述された XML ファイルです。SMF マニフェストの作成の詳細については、[Oracle Solaris 10 システム管理者ドキュメント \(http://download.oracle.com/docs/cd/E18752_01/index.html\)](http://download.oracle.com/docs/cd/E18752_01/index.html)を参照してください。

注記 - コンソールを使用してゲストドメインの英語版以外の OS にアクセスするには、コンソールの端末が、その OS が必要とするロケールになっている必要があります。

コンソールグループの使用

仮想ネットワーク端末サーバーデーモン (`vntsd`) により、1 つの TCP ポートを使用して複数のドメインコンソールにアクセスできます。Logical Domains Manager は、ドメインの作成時に、そのドメインのコンソール用の新しいデフォルトグループを作成することにより、各コンソールに一

意の TCP ポートを割り当てます。TCP ポートは、コンソール自体ではなくコンソールグループに割り当てられます。コンソールは、`set-vcons` サブコマンドを使用して既存のグループにバインドできます。

▼ 複数のコンソールを 1 つのグループにまとめる方法

1. ドメインのコンソールを 1 つのグループにバインドします。

次の例では、3 つの異なるドメイン (`ldg1`、`ldg2`、`ldg3`) のコンソールを同じコンソールグループ (`group1`) にバインドします。

```
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg1
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg2
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg3
```

2. 関連付けられた TCP ポート (この例ではポート 5000 の `localhost`) に接続します。

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
```

いずれかのドメインコンソールの選択を求めるプロンプトが表示されます。

3. `l` (`list`) を選択して、グループ内のドメインを一覧表示します。

```
primary-vnts-group1: h, l, c{id}, n{name}, q: l
DOMAIN ID          DOMAIN NAME          DOMAIN STATE
0                   ldg1                 online
1                   ldg2                 online
2                   ldg3                 online
```

注記 - コンソールを別のグループまたは `vcc` インスタンスに再度割り当てるには、ドメインがバインドされていない状態、つまり、アクティブでない状態である必要があります。SMF を構成および使用して `vntsd` を管理する方法、およびコンソールグループの使用の詳細は、`vntsd(1M)` のマニュアルページを参照してください。

負荷が大きいドメインの停止処理がタイムアウトする可能性

`ldm stop-domain` コマンドは、ドメインが完全に停止する前にタイムアウトする可能性があります。このような状況が発生すると、Logical Domains Manager によって次のようなエラーが返されます。

```
LDom ldg8 stop notification failed
```

しかし、ドメインが停止要求をまだ処理している可能性があります。`ldm list-domain` コマンドを使用して、ドメインのステータスを確認します。たとえば、次のように表示されます。

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU MEMORY  UTIL UPTIME
ldg8          active s---- 5000   22  3328M  0.3% 1d 14h 31m
```

前述のリストには、ドメインがアクティブと表示されていますが、`s` フラグはドメインが停止処理中であることを示しています。これは、一時的な状態であるはずです。

次の例は、ドメインがすでに停止していることを示しています。

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU MEMORY  UTIL UPTIME
ldg8          bound  ----- 5000   22  3328M
```

`ldm stop` コマンドは `shutdown` コマンドを使用してドメインを停止します。通常、シャットダウンシーケンスの実行には、`ldm stop -q` コマンドの実行によるクイック停止よりも大幅に時間がかかります。[ldm\(1M\)](#) マニュアルページを参照してください。

シャットダウンシーケンスが長時間になると、次のようなタイムアウトメッセージが生成される場合があります。

```
domain-name stop timed out. The domain might still be in the process of shutting down.
Either let it continue, or specify -f to force it to stop.
```

このシャットダウンシーケンスの実行中は、ドメインに `s` フラグも表示されます。

Oracle VM Server for SPARC による Oracle Solaris OS の運用

このセクションでは、Logical Domains Manager によって作成された構成のインスタンスが作成されたあとで、Oracle Solaris OS を使用する際の動作がどのように変化するかについて説明します。

Oracle Solaris OS の起動後には OpenBoot ファームウェアを使用できない

Oracle Solaris OS の起動後には、OpenBoot ファームウェアは使用できません。これは、OpenBoot ファームウェアがメモリーから削除されるためです。

Oracle Solaris OS から `ok` プロンプトを表示するには、Oracle Solaris OS `halt` コマンドを使用してドメインを停止する必要があります。

サーバーの電源再投入の実行

Oracle VM Server for SPARC ソフトウェアを実行しているシステムで、サーバーの電源の再投入が必要になる保守を行う場合は、必ず最初に現在の論理ドメイン構成を SP に保存する必要があります。

現在のドメイン構成を SP に保存するには、次のコマンドを使用します。

```
# ldm add-config config-name
```

Oracle Solaris OS ブレークの結果

Oracle Solaris OS ブレークは次のようにして開始できます。

1. 入力デバイスが `keyboard` に設定されているときに、L1-A キーシーケンスを押した場合。
2. 仮想コンソールが `telnet` プロンプトにあるときに、`send break` コマンドを入力した場合。

このようなブレークを開始すると、Oracle Solaris OS は次のプロンプトを発行します。

```
c)ontinue, s)ync, r)eset, h)alt?
```

このようなタイプのブレークが発生したあとのシステムの動作を表す文字を入力します。

制御ドメインのリブート結果

`reboot` および `shutdown -i 5` コマンドを使用して、制御 (primary) ドメインをリブートできます。

■ reboot:

- ほかのドメインが構成されていない場合。正常な停止を行わずに制御ドメインをリブートします。電源は切られません。
- ほかのドメインが構成されている場合。正常な停止を行わずに制御ドメインをリブートします。電源は切られません。

■ shutdown -i 5:

- ほかのドメインが構成されていない場合。正常な停止後にホストの電源が切られ、SP で電源が投入されるまで切られたままとなります。
- ほかのドメインが構成されている場合。適切に停止してリブートします。電源は切られません。

ルートドメインの役割を持つドメインをリブートした結果については、[154 ページの「PCIe エンドポイントを構成した状態のルートドメインのリブート」](#)を参照してください。

サービスプロセッサでの Oracle VM Server for SPARC の使用

このセクションでは、Logical Domains Manager で Integrated Lights Out Manager (ILOM) サービスプロセッサ (Service Processor, SP) を使用するときに関する情報について説明します。ILOM ソフトウェアの使用法については、使用しているプラットフォーム固有のドキュメント (<http://www.oracle.com/technetwork/documentation/sparc-tseries-servers-252697.html>) を参照してください。

既存の ILOM コマンドに、`config` オプションを 1 つ追加できます。

```
-> set /HOST/bootmode config=config-name
```

このオプションを使用すると、次の電源投入時の構成を出荷時構成 (`factory-default`) などの別の構成に設定できます。

ホストの電源が投入されているか切断されているかにかかわらず、このコマンドを実行できます。次のホストリセットまたは電源投入時に有効になります。

論理ドメイン構成をリセットするには、このオプションを `factory-default` に設定します。

```
-> set /HOST/bootmode config=factory-default
```

また、`ldm add-config` コマンドを使用して Logical Domains Manager で作成され、サービスプロセッサ (SP) に保存されているほかの構成を選択することもできます。Logical Domains Manager の `ldm add-config` コマンドで指定した名前を使用して、ILOM の `bootmode` コマンドでその構成を選択できます。たとえば、`ldm-config1` という名前の構成が保存されているとすると、次のように指定します。

```
-> set /HOST/bootmode config=ldm-config1
```

ここで、システムの電源再投入を実行して、新しい構成をロードする必要があります。

`ldm add-config` コマンドの詳細は、[ldm\(1M\)](#) マニュアルページを参照してください。

ドメインの依存関係の構成

Logical Domains Manager を使用して、ドメイン間の依存関係を確立できます。依存する 1 つ以上のドメインを持つドメインは、マスタードメインと呼ばれます。別のドメインに依存するドメインは、スレーブドメインと呼ばれます。

master プロパティを設定することによって、各スレーブドメインに最大 4 つのマスタードメインを指定できます。たとえば、次に示すコマンドで区切られたリストでは、pine スレーブドメインに 4 つのマスタードメインを指定しています。

```
# ldm add-domain master=alpha,beta,gamma,delta pine
```

alpha, beta, gamma, および delta マスタードメインはすべて、stop の失敗ポリシーを指定します。

各マスタードメインには、マスタードメインに障害が発生した場合のスレーブドメインの動作を指定できます。たとえば、マスタードメインに障害が発生した場合、そのスレーブドメインでパニックを発生させる必要があることがあります。スレーブドメインに複数のマスタードメインがある場合は、各マスタードメインが同じ失敗ポリシーを持つ必要があります。そのため、失敗した最初のマスタードメインは、定義済みの失敗ポリシーをそのすべてのスレーブドメインでトリガーします。

マスタードメインの障害ポリシーは、failure-policy プロパティに次のいずれかの値を設定することによって制御できます。

- ignore は、すべてのスレーブドメインを無視します
- panic は、すべてのスレーブドメインにパニックを発生させます (ldm panic コマンドを実行することと似ています)
- reset は、すべてのスレーブドメインをただちに停止して再起動します (ldm stop -f コマンド、ldm start コマンドの順に実行することと似ています)
- stop は、すべてのスレーブドメインを停止します (ldm stop -f コマンドを実行することと似ています)

この例では、マスタードメインの障害ポリシーが次のように指定されています。

```
primary# ldm set-domain failure-policy=ignore apple
primary# ldm set-domain failure-policy=panic lemon
primary# ldm set-domain failure-policy=reset orange
primary# ldm set-domain failure-policy=stop peach
primary# ldm set-domain failure-policy=stop alpha
primary# ldm set-domain failure-policy=stop beta
primary# ldm set-domain failure-policy=stop gamma
primary# ldm set-domain failure-policy=stop delta
```

このメカニズムを使用して、ドメイン間の明示的な依存関係を作成できます。たとえば、ゲストドメインが、サービスドメインに暗黙に依存し、その仮想デバイスを提供しているとします。ゲストドメインが依存しているサービスドメインが実行されていない場合、ゲストドメインの入出力はブロックされます。ゲストドメインをサービスドメインのスレーブドメインとして定義することによって、サービスドメインが停止した場合のゲストドメインの動作を指定できます。このような依存関係が確立されていない場合、ゲストドメインはサービスドメインが使用可能になるのを待機します。

注記 - Logical Domains Manager では、依存サイクルを作成するようなドメインの依存関係は作成できません。詳細は、[409 ページの「依存サイクル」](#)を参照してください。

ドメインの依存関係の XML の例は、[例23-6「ldom_info の XML 出力の例」](#)を参照してください。

ドメインの依存関係の例

次の例は、ドメインの依存関係を構成する方法を示します。

例 17-1 ドメインの依存関係を使用した障害ポリシーの構成

最初のコマンドは、twizzle というマスタートドメインを作成します。このコマンドは、`failure-policy=reset` を使用して、twizzle ドメインに障害が発生した場合にスレーブドメインをリセットするように指定します。2 つめのコマンドは、primary というマスタートドメインに変更を加えます。このコマンドは、`failure-policy=reset` を使用して、primary ドメインに障害が発生した場合にスレーブドメインをリセットするように指定します。3 つめのコマンドは、2 つのマスタートドメイン twizzle と primary に依存する、chocktaw というスレーブドメインを作成します。このスレーブドメインは、`master=twizzle,primary` を使用して、マスタートドメインを指定します。twizzle または primary のいずれかのドメインに障害が発生した場合、chocktaw ドメインはリセットされます。

```
primary# ldm add-domain failure-policy=reset twizzle
primary# ldm set-domain failure-policy=reset primary
primary# ldm add-domain master=twizzle,primary chocktaw
```

例 17-2 マスタートドメインを割り当てるためのドメインの変更

この例は、`ldm set-domain` コマンドを使用して orange ドメインに変更を加え、primary をマスタートドメインとして割り当てます。2 つめのコマンドは、`ldm set-domain` コマンドを使用し

て、orange および primary を tangerine ドメインのマスタードメインとして割り当てます。3 つめのコマンドは、これらすべてのドメインに関する情報を一覧表示します。

```
primary# ldm set-domain master=primary orange
primary# ldm set-domain master=orange,primary tangerine
primary# ldm list -o domain
```

NAME	STATE	FLAGS	UTIL
primary	active	-n-cv-	0.2%

SOFTSTATE
Solaris running

HOSTID
0x83d8b31c

CONTROL
failure-policy=ignore

DEPENDENCY
master=

```
-----
NAME          STATE      FLAGS    UTIL
orange        bound     -----
```

HOSTID
0x84fb28ef

CONTROL
failure-policy=ignore

DEPENDENCY
master=primary

```
-----
NAME          STATE      FLAGS    UTIL
tangerine     bound     -----
```

HOSTID
0x84f948e9

CONTROL
failure-policy=ignore

DEPENDENCY
master=orange,primary

例 17-3 解析可能なドメインリストの表示

次に、解析可能な出力を使用した一覧表示の例を示します。

```
primary# ldm list -o domain -p
```


依存サイクル

Logical Domains Manager では、依存サイクルを作成するようなドメインの依存関係は作成できません。依存サイクルとは、スレーブドメインが自身に依存したり、マスタードメインがそのスレーブドメインのいずれかに依存したりすることになる、2 つ以上のドメイン間の関係です。

Logical Domains Manager は、依存関係を追加する前に、依存サイクルが存在するかどうかを確認します。Logical Domains Manager は、スレーブドメインから検索を開始し、マスターアレイによって指定されているすべてのパスに沿って、パスの最後に到達するまで検索を行います。途中で依存サイクルが見つかったら、エラーとして報告されます。

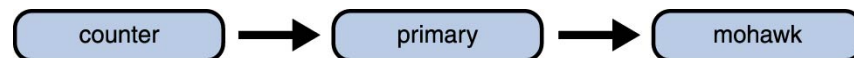
次の例は、依存サイクルがどのように作成されるかを示します。最初のコマンドは、`mohawk` というスレーブドメインを作成します。このドメインは、マスタードメインに `primary` を指定します。その結果、`mohawk` は、次の図に示すような依存関係の連鎖で `primary` に依存します。

図 17-1 単一のドメインの依存関係



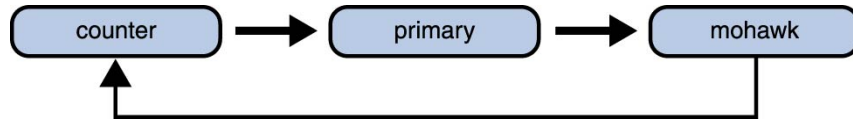
2 つめのコマンドは、`primary` というスレーブドメインを作成します。このドメインは、マスタードメインに `counter` を指定します。その結果、次の図に示すような依存関係の連鎖で、`mohawk` が `primary` に依存し、`primary` が `counter` に依存します。

図 17-2 複数のドメインの依存関係



3 つめのコマンドは、`counter` ドメインと `mohawk` ドメインとの間に依存関係の作成を試みます。これによって、次の図に示すような依存サイクルが生成されます。

図 17-3 ドメインの依存サイクル



次のエラーメッセージが表示されて `ldm set-domain` コマンドが失敗します。

```
# ldm add-domain master=primary mohawk
# ldm set-domain master=counter primary
# ldm set-domain master=mohawk counter
Dependency cycle detected: LDom "counter" indicates "primary" as its master
```

CPU およびメモリアドレスのマッピングによるエラー発生箇所の確認

このセクションでは、Oracle Solaris 障害管理アーキテクチャー (FMA) によって報告された情報と、障害としてマークされている論理ドメインリソースを相互に関連付ける方法について説明します。

FMA では、物理 CPU 番号に関する CPU エラーと、物理メモリアドレスに関するメモリエラーを報告します。

エラーが発生した論理ドメインと、そのドメイン内の対応する仮想 CPU 番号または実メモリアドレスを確認する場合は、マッピングを実行する必要があります。

CPU マッピング

特定の物理 CPU 番号に対応しているドメインとそのドメイン内の仮想 CPU 番号を調べることができます。

最初に、次のコマンドを使用してすべてのドメインの解析可能な長形式のリストを生成します。

```
primary# ldm list -l -p
```

リストの **VCPU** セクションで、物理 CPU 番号に等しい **pid** フィールドを持つエントリを探します。

- このようなエントリが見つかった場合、CPU はそのエントリが表示されたドメインに存在し、そのドメイン内の仮想 CPU 番号がエントリの **vid** フィールドに指定されています。
- このようなエントリが見つからない場合、CPU はどのドメインにも存在しません。

メモリーのマッピング

特定の物理メモリーアドレス (PA) に対応しているドメインとそのドメイン内の実メモリーアドレスを調べることができます。

最初に、すべてのドメインの解析可能な長形式のリストを生成します。

```
primary# ldm list -l -p
```

リストの **MEMORY** セクションの行を検索します。この場合、PA は pa から $(pa + size - 1)$ の包括範囲内にあります。つまり、 $pa \leq PA \leq (pa + size - 1)$ です。 pa および $size$ は、行で対応するフィールドの値を指します。

- このようなエントリが見つかった場合、PA はそのエントリが表示されたドメインに存在し、そのドメイン内の対応する実アドレスが $ra + (PA - pa)$ によって求められます。
- このようなエントリが見つからない場合、PA はどのドメインにも存在しません。

CPU およびメモリーのマッピングの例

例 17-4 ドメイン構成の特定

次のコマンドは、論理ドメイン構成の解析可能な長形式のリストを生成します。

```
primary# ldm list -l -p
VERSION 1.6
DOMAIN|name=primary|state=active|flags=normal,control,vio-service|
cons=SP|ncpu=4|mem=1073741824|util=0.6|uptime=64801|
softstate=Solaris running
VCPU
|vid=0|pid=0|util=0.9|strand=100
|vid=1|pid=1|util=0.5|strand=100
|vid=2|pid=2|util=0.6|strand=100
|vid=3|pid=3|util=0.6|strand=100
```

```

MEMORY
|ra=0x80000000|pa=0x80000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
|dev=pci@7c0|alias=bus_b
...
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|
ncpu=2|mem=805306368|util=29|uptime=903|
softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x80000000|pa=0x48000000|size=805306368
...
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|
ncpu=3|mem=1073741824|util=35|uptime=775|
softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x80000000|pa=0x78000000|size=1073741824
...

```

例 17-5 物理 CPU 番号に対応する仮想 CPU の特定

論理ドメイン構成は、[例17-4「ドメイン構成の特定」](#)に示されています。この例では、物理 CPU 番号が 5 であるドメインと仮想 CPU、および物理アドレス 0x7e816000 に対応するドメインおよび実アドレスを特定する方法について説明します。

リストで pid フィールドが 5 である VCPU エントリを探すと、論理ドメイン ldg1 の下に次のエントリが見つかります。

```
|vid=1|pid=5|util=29|strand=100
```

したがって、物理 CPU 番号 5 はドメイン ldg1 に存在し、そのドメイン内には仮想 CPU 番号 1 があります。

リストの MEMORY エントリを探すと、ドメイン ldg2 の下に次のエントリが見つかります。

```
ra=0x80000000|pa=0x78000000|size=1073741824
```

この場合、 $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$ 、つまり、 $pa \leq PA \leq (pa + size - 1)$ となります。したがって、PA はドメイン ldg2 にあり、対応する実アドレスは $0x80000000 + (0x7e816000 - 0x78000000) = 0xe816000$ です。

ユニバーサル固有識別子の使用

各ドメインにはユニバーサル固有識別子 (UUID) が割り当てられます。UUID は、ドメインの作成時に割り当てられます。レガシードメインについては、ldmd デーモンの初期化時に UUID が割り当てられます。

注記 - `ldm migrate-domain -f` コマンドを使用して、古いバージョンの Logical Domains Manager を実行しているターゲットマシンにドメインを移行した場合、UUID は消失します。古いバージョンの Logical Domains Manager を実行しているソースマシンからドメインを移行すると、移行の一環として、そのドメインに新しい UUID が割り当てられます。それ以外の場合、UUID は移行されます。

`ldm list -l`、`ldm list-bindings`、または `ldm list -o domain` コマンドを実行すると、ドメインの UUID を取得できます。次の例は、ldg1 ドメインの UUID を示しています。

```
primary# ldm create ldg1
primary# ldm ls -l ldg1
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
ldg1          inactive  -----
UUID
6c908858-12ef-e520-9eb3-f1cd3dbc3a59

primary# ldm ls -l -p ldg1
VERSION 1.6
DOMAIN|name=ldg1|state=inactive|flags=|cons=|ncpu=|mem=|util=|uptime=
UUID|uuid=6c908858-12ef-e520-9eb3-f1cd3dbc3a59
```

仮想ドメイン情報コマンドと API

`virtinfo` コマンドを使用して、実行中の仮想ドメインに関する情報を収集することができます。また、仮想ドメイン情報 API を使用して、仮想ドメインに関する情報を収集するプログラムを作成することもできます。

コマンドまたは API を使用して、仮想ドメインについて収集できる情報の一覧を、次に示します。

- ドメインの種類 (実装、制御、ゲスト、I/O、サービス、ルート)
- 仮想ドメインマネージャーにより決定されるドメイン名
- ドメインのユニバーサル固有識別子 (UUID)

- ドメインの制御ドメインのネットワークノード名
- ドメインが実行されているシャーシのシリアル番号

virtinfo コマンドについては、virtinfo(1M) のマニュアルページを参照してください。API については、libv12n(3LIB) および v12n(3EXT) のマニュアルページを参照してください。

論理ドメインチャネルの使用

Oracle VM Server for SPARC では、論理ドメインチャネル (LDC) を使用して、すべての通信 (コンソール、仮想 I/O、制御トラフィックなど) を実装します。LDC は、2 つのエンドポイント間の通信を可能にするために使用される方法です。通常、各エンドポイントは異なるドメインにありますが、それらのエンドポイントを同じドメインに配置してループバック通信を可能にすることができます。

このソフトウェアおよびシステムファームウェアによって、制御ドメインやゲストドメインに使用できる LDC エンドポイントの大きなプールが提供されます。この LDC エンドポイントプールは、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、SPARC M7 シリーズサーバー、および Fujitsu M10 サーバーでのみ使用できます。プール内の LDC の数は、次のようにプラットフォームタイプに基づいています。

- **SPARC T4** サーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、合計 98304 個の LDC エンドポイント
- **SPARC T5** サーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、合計 98304 個の LDC エンドポイント
- **SPARC T7** シリーズサーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、合計 98304 個の LDC エンドポイント
- **SPARC M5** サーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、物理ドメインあたり 98304 個の LDC エンドポイント
- **SPARC M6** サーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、物理ドメインあたり 98304 個の LDC エンドポイント
- **SPARC M7** シリーズサーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、物理ドメインあたり 98304 個の LDC エンドポイント
- **Fujitsu M10** サーバー – ゲストドメインあたり 1984 個の LDC エンドポイント、合計 196608 個の LDC エンドポイント

LDC エンドポイントプールをサポートするために必要なシステムファームウェアは、SPARC T4 サーバーの場合は 8.5.2、SPARC T5 サーバー、SPARC M5 サーバー、および SPARC M6 サーバーの場合は 9.2.1、SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーの場合は 9.4.3、Fujitsu M10 サーバーの場合は XCP2240 です。

サポートされるプラットフォームや UltraSPARC T2、UltraSPARC T2 Plus、または SPARC T3 プラットフォーム上で古いバージョンのシステムファームウェアを実行している場合は、次の LDC エンドポイントの制限が引き続き適用されます。

- **UltraSPARC T2** サーバー – 512 個の LDC エンドポイント
- **UltraSPARC T2 Plus** サーバー – 768 個の LDC エンドポイント
- **SPARC T3** サーバー – 768 個の LDC エンドポイント
- **SPARC T4** サーバー – 768 個の LDC エンドポイント
- **SPARC T5** サーバー – 768 個の LDC エンドポイント
- **SPARC T7** シリーズサーバー – 768 個の LDC エンドポイント
- **SPARC M5** サーバー – 768 個の LDC エンドポイント
- **SPARC M6** サーバー – 768 個の LDC エンドポイント
- **SPARC M7** シリーズサーバー – 768 個の LDC エンドポイント
- **Fujitsu M10** サーバー – 768 個の LDC エンドポイント

この制限は制御ドメインで問題になることがあります。これは、仮想 I/O データ通信と、Logical Domains Manager によるほかのドメイン制御の両方に多数の LDC エンドポイントが使用される可能性があるからです。

サービスの追加やドメインのバインドを試みたために、LDC エンドポイントの数が 1 つのドメインでの制限を超えてしまった場合、その操作は失敗して、次のようなエラーメッセージが表示されます。

```
13 additional LDCs are required on guest primary to meet this request,
but only 9 LDCs are available
```

次のガイドラインを使用すると、LDC エンドポイントの使用計画を正しく立てたり、制御ドメインで LDC 機能のオーバーフローが発生する可能性のある原因を説明したりできます。

- 制御ドメインでは、構成されているほかのドメインの数に関係なく、ハイパーバイザ、障害管理アーキテクチャー (FMA)、およびシステムプロセッサ (SP) とのさまざまな通信目的で、約 15 個の LDC エンドポイントを使用します。制御ドメインによって使用される LDC エンドポイントの数は、プラットフォーム、および使用されているソフトウェアのバージョンによって異なります。
- Logical Domains Manager は、ドメイン (自身を含む) ごとに制御トラフィック用の LDC エンドポイントを制御ドメインに 1 つ割り当てます。

- 制御ドメインの各仮想 I/O サービスは、そのサービスに接続されているクライアントごとに LDC エンドポイントを 1 つ使用します。各ドメインには、仮想ネットワーク、仮想ディスク、および仮想コンソールが少なくとも 1 つずつ必要です。

次の式では、これらのガイドラインを取り入れて、制御ドメインで必要とされる LDC エンドポイントの数を算出します。

$$15 + \text{number-of-domains} + (\text{number-of-domains} \times \text{number-of-virtual-services}) = \text{total-LDC-endpoints}$$

number-of-domains は制御ドメインを含むドメインの総数であり、*number-of-virtual-services* はこのドメインによって処理される仮想 I/O デバイスの総数です。

次の例は、制御ドメインが 1 つ、追加のドメインが 8 つある場合に、式を使用して LDC エンドポイントの数を算出する方法を示しています。

$$15 + 9 + (8 \times 3) = 48 \text{ LDC エンドポイント}$$

次の例では、ゲストドメインが 45 あり、各ドメインに 5 つの仮想ディスク、2 つの仮想ネットワーク、および 1 つの仮想コンソールがあります。計算によって次の結果が得られます。

$$15 + 46 + 45 \times 8 = 421 \text{ LDC エンドポイント}$$

使用しているプラットフォームでサポートされている LDC エンドポイントの数に応じて、Logical Domains Manager が構成を受け入れるか拒否するかが決まります。

制御ドメインで LDC エンドポイントを使い果たした場合は、サービルドメインまたは I/O ドメインを作成して仮想 I/O サービスをゲストドメインに提供することを検討してください。このアクションを実行すると、制御ドメインではなく、I/O ドメインとサービルドメインに LDC エンドポイントを作成できます。

ゲストドメインでも、LDC エンドポイントを使い果たす可能性があります。このような状況は、`inter-vnet-link` プロパティが `on` に設定されることで、相互に直接接続するゲストドメインに追加の LDC エンドポイントが割り当てられるために起こることがあります。

次の式では、`inter-vnet-link=off` のときにゲストドメインで必要とされる LDC エンドポイントの数を算出します。

$$2 + \text{number-of-vnets} + \text{number-of-vdisks} = \text{total-LDC-endpoints}$$

2 は仮想コンソールと制御トラフィックを表し、*number-of-vnets* はゲストドメインに割り当てられている仮想ネットワークデバイスの総数、*number-of-vdisks* はゲストドメインに割り当てられている仮想ディスクの総数です。

次の例は、`inter-vnet-link=off` で、仮想ディスクと仮想ネットワークが 2 つずつある場合に、式を使用して 1 ゲストドメインあたりの LDC エンドポイント数を算出する方法を示しています。

$2 + 2 + 2 = 6$ LDC エンドポイント

次の式では、`inter-vnet-link=on` のときにゲストドメインで必要とされる LDC エンドポイントの数を算出します。

$$2 + [[(\text{number-of-vnets-from-vswX} \times \text{number-of-vnets-in-vswX})] \dots] + \text{number-of-vdisks} = \text{total-LDC-endpoints}$$

2 は仮想コンソールと制御トラフィックを表し、`number-of-vnets-from-vswX` は `vswX` 仮想スイッチからゲストドメインに割り当てられている仮想ネットワークデバイスの総数、`number-of-vnets-in-vswX` は `vswX` 仮想スイッチ上の仮想ネットワークデバイスの総数、`number-of-virtual-disks` はゲストドメインに割り当てられている仮想ディスクの総数です。

次の例は、`inter-vnet-link=on` で、仮想ディスクと仮想スイッチが 2 つずつある場合に、式を使用して 1 ゲストドメインあたりの LDC エンドポイント数を算出する方法を示しています。最初の仮想スイッチには 8 つの仮想ネットワークがあり、そのうちの 4 つがドメインに割り当てられます。2 つめの仮想スイッチは、8 つすべての仮想ネットワークをドメインに割り当てます。

$2 + (4 \times 8) + (8 \times 8) + 2 = 100$ LDC エンドポイント

ゲストドメインで LDC エンドポイントが不足する問題を解決するには、`ldm add-vsw` または `ldm set-vsw` コマンドを使用して `inter-vnet-link` プロパティを `off` に設定することを確認してください。このアクションによって、そのドメイン、または仮想ネットワークデバイスが存在するドメインの LDC エンドポイントの数が減ります。ただし、`off` プロパティ値は、仮想スイッチを含むサービスドメインには影響しません (サービスドメインでは各仮想ネットワークデバイスへの LDC 接続を引き続き必要とするため)。このプロパティが `off` に設定されていると、LDC チャネルは `inter-vnet` 通信に使用されません。代わりに、LDC チャネルは仮想ネットワークデバイスと仮想スイッチデバイス間の通信のみに割り当てられます。[ldm\(1M\)](#) マニュアルページを参照してください。

注記 - `inter-vnet` リンクの割り当てを無効にすると、LDC エンドポイントの数が減りますが、ゲスト間のネットワークパフォーマンスに悪影響を及ぼす可能性があります。このパフォーマンス低下が生じるのは、ゲスト間のすべての通信トラフィックが、あるゲストドメインから別のゲストドメインに直接移動するのではなく、仮想スイッチを経由するからです。

多数のドメインのブート

使用しているサーバーに応じて、次の数のドメインをブートできます。

- 物理的なパーティションあたり で最大 256
- 物理ドメインあたり SPARC M7 シリーズサーバーで最大 128
- 物理ドメインあたり SPARC M6 サーバーで最大 128
- 物理ドメインあたり SPARC M5 サーバーで最大 128
- SPARC T7 シリーズサーバーで最大 128
- SPARC T5 サーバーで最大 128
- SPARC T4 サーバーで最大 128
- SPARC T3 サーバーで最大 128
- UltraSPARC T2 Plus サーバーで最大 128
- UltraSPARC T2 Plus サーバーで最大 64

未割り当ての仮想 CPU を使用できる場合は、それらをサービスドメインに割り当てて、仮想 I/O リクエストの処理に役立てます。32 を超えるドメインを作成する場合は、サービスドメインに 4 - 8 個の仮想 CPU を割り当ててください。ドメインの最大構成ではサービスドメインに CPU が 1 つしか含まれない場合、ドメインを構成して使用する際は、この単一の CPU に余分な負荷を与えないようにします。仮想スイッチ (vsw) サービスは、マシン内で使用可能なすべてのネットワークアダプタに分散させてください。たとえば、Sun SPARC Enterprise T5240 サーバー上で 128 ドメインをブートする場合は、4 つの vsw サービスを作成し、各サービスに 32 の仮想ネットワーク (vnet) インスタンスを割り当てます。vsw サービスあたり 32 を超える vnet インスタンスを割り当てると、サービスドメインでハードハングが発生することがあります。

最大構成を実行するには、ゲストドメインをサポートするために適切な量のメモリがマシンに必要です。このメモリ量は、使用しているプラットフォームと OS によって異なります。『[Oracle Solaris 10 8/11 Installation Guide: Planning for Installation and Upgrade](#)』および『[Installing Oracle Solaris 11.3 Systems](#)』のうち、使用しているプラットフォームのドキュメントを参照してください。

ドメインで使用されている vsw サービスが複数のドメイン内の多数の仮想ネットワークにサービスを提供していると、ゲストドメイン内のメモリやスワップ領域の使用量が増加します。この増加は、vsw に接続されているすべての vnet インスタンス間のピアツーピアリンクによるものです。サービスドメインでは、追加のメモリを設定すると役立ちます。64 を超えるドメインを実行する場合の推奨される最小サイズは 4G バイトです。10 以下のドメインをグループにまとめて

起動し、ブートが完了するまで待機してから次のグループを起動します。この方法は、ドメインにオペレーティングシステムをインストールする際にも適用されます。リンクの数は、inter-vnet リンクを無効にすることで減らすことができます。[Inter-Vnet LDC チャンネル](#)を参照してください。

Oracle VM Server for SPARC システムの正常な停止と電源の再投入

構成を SC に保存したあとで構成に変更を加えた場合は、Oracle VM Server for SPARC システムの電源を切って再投入する前に、保持する必要がある最新の構成を必ず保存してください。

▼ アクティブなドメインが複数存在するシステムの電源を切る方法

1. I/O ドメイン以外のすべてのドメインをシャットダウンし、停止して、バインドを解除します。
2. アクティブな I/O ドメインをすべてシャットダウンし、停止して、バインドを解除します。
3. ドメインを init 状態 5 に変更します。

```
primary# shutdown -i 5
```

shutdown コマンドを使用する代わりに、init 5 コマンドを使用することもできます。

▼ システムの電源を再投入する方法

1. システムの電源を切ります。
[419 ページの「アクティブなドメインが複数存在するシステムの電源を切る方法」](#)を参照してください。
2. SP を使用してシステムの電源を入れます。

Logical Domains 変数の永続性

変数の更新はリブートの場合には保持されますが、電源を再投入した場合には、制御ドメインの OpenBoot ファームウェアから開始されるか、変数の更新後に SC に構成が保存されていないかぎり保持されません。

次のような状態に注意してください。

- バインドされているゲストドメインが存在せず、遅延再構成が進行中でない場合に制御ドメインがリブートすると、SC はシステムの電源の再投入を実行します。
- ゲストドメインがバインドされているか、アクティブな場合 (または制御ドメインで遅延再構成が進行中の場合) に制御ドメインがリブートすると、SC はシステムの電源の再投入を実行しません。

ドメインの Logical Domains 変数は、次のいずれかの方法で指定できます。

- OpenBoot プロンプトを使用する。
- Oracle Solaris OS の `eeeprom(1M)` コマンドを使用する。
- Logical Domains Manager CLI (`ldm`) を使用する。
- 制限された方法で、システムコントローラ (SC) から `bootmode` コマンドを使用する。この方法は、特定の変数が `factory-default` 構成に存在する場合にのみ、それらの変数でのみ使用できます。

これらのいずれかの方法を使用して行われた変数の更新が、ドメインをリブートしても必ず保持されるようにしてください。また、それらの変数の更新はこのあと SC に保存されたすべてのドメイン構成に必ず適用されます。

Oracle VM Server for SPARC 3.3 ソフトウェアでは、更新された変数が期待どおりに保持されない場合があります。

- どの方法で変数を更新した場合でも、変数はそのドメインをリブートしても保持されます。ただし、システムの電源を再投入すると、新しい論理ドメイン構成が SC に保存されていないかぎり、それらは保持されません。

ただし、制御ドメインでは、OpenBoot ファームウェアのコマンドまたは `eeeprom` コマンドのどちらかを使用して行われた更新は、そのあとに新しい論理ドメイン構成を SC に保存しなくても、システムの電源を再投入したあとも保持されます。`eeeprom` コマンドは、この動作を SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、および SPARC M7 シリーズサーバーと、少なくともバージョン 8.2.1 のシステムファームウェアを実行している SPARC T3 サーバーおよび SPARC T4 サーバー上でサポートします。

Logical Domains 変数の変更について懸念がある場合は、次のいずれかの手順を実行します。

- システムで `ok` プロンプトを表示し、変数を更新します。
- Logical Domains Manager を無効にして変数を更新します。

```
# svcadm disable ldmd
update variables
# svcadm enable ldmd
```

- Live Upgrade を実行し、次のコマンドを実行します。

```
# svcadm disable -t ldmd
# luactivate be3
# init 6
```

`ntptime` コマンドなどを使用して論理ドメイン上の時間または日付を変更すると、その変更はドメインをリブートしても保持されますが、ホストの電源を再投入した場合は保持されません。時間の変更を保持するには、時間が変更された構成を SP に保存して、この構成からブートします。

これらの問題を解決するために、15375997、15387338、15387606、および 15415199 のバグ ID が報告されています。

割り込み制限の調整

ハードウェアは無限の数の割り込みを提供するため、Oracle Solaris は各デバイスが使用できる割り込みの数を制限します。デフォルトの制限は一般的なシステム構成のニーズに一致する必要がありますが、特定のシステム構成ではこの値の調整が必要になることがあります。

注記 - これらの制限は、SPARC M7 シリーズサーバーおよび SPARC T7 シリーズサーバーには適用されません。

PCIe バスで I/O 仮想化を有効にすると、割り込みハードウェアリソースが各 I/O ドメインに割り当てられます。各ドメインは有限の数のこれらのリソースに割り当てられ、何らかの割り込み割り当ての問題につながる場合があります。この状況は、UltraSPARC T2、UltraSPARC T2 Plus、SPARC T3、SPARC T4、SPARC T5、SPARC M5、および SPARC M6 プラットフォームにのみ影響します。

Oracle Solaris コンソール上のこの警告は、I/O デバイスドライバの接続中に、供給された割り込み数が使い果たされたことを意味します。

```
WARNING: ddi_intr_alloc: cannot fit into interrupt pool
```

特に、システムが複数の論理ドメインにパーティション化されている場合や、任意のゲストドメインに割り当てられる I/O デバイスの数が多すぎる場合は、この制限に調整が必要になることがあります。Oracle VM Server for SPARC では、全割り込み数を小さいセットに分けて、ゲストドメインに割り当てます。ゲストドメインに割り当てられる I/O デバイスが多すぎる場合、その割り込み供給が少なすぎるため、各デバイスにデフォルトの割り込み数を提供できないことがあります。そのため、ゲストドメインは、すべてのドライバを完全に接続する前にその割り込み供給量を使い果たしてしまいます。

ドライバの中には、その割り込み数を Oracle Solaris OS で自動的に調整できるようにするオプションのコールバックルーチンを提供するものもあります。このようなドライバにはデフォルトの制限が適用されません。

`::irmpools` および `::irmreqs` MDB マクロを使用して、割り込みがどのように使用されているかを調べます。`::irmpools` マクロは、プールに分けられた割り込みの総供給量を表示します。`::irmreqs` マクロは、各プールにマップされているデバイスを表示します。デバイスごとに、`::irmreqs` は、オプションのコールバックルーチンによってデフォルトの制限が強制されるかどうか、各ドライバがリクエストした割り込みの数、および各ドライバの割り込みの数を表示します。

これらのマクロは接続に失敗したドライバの情報を表示しませんが、その情報を使用すると、デフォルトの制限を調整できる範囲を計算できます。デフォルトの制限を調整することによって、コールバックルーチンを提供せずに複数の割り込みを使用するデバイスが、より少ない数の割り込みを使用するように強制できます。このようなデバイスについては、デフォルトの制限を小さくして、ほかのデバイスで利用できる割り込みを解放します。

デフォルトの制限を調整するには、`/etc/system` ファイル内の `ddi_msix_alloc_limit` プロパティを 1-8 の値に設定します。次に、その変更が反映されるようにシステムをリブートします。

`/etc/system` プロパティ値の正しい作成または更新については、[400 ページの「/etc/system ファイルのプロパティ値の更新」](#)を参照してください。

パフォーマンスを最大にするには、開始時に大きな値を割り当て、システムが警告なしで正常にブートするまで、それらの値を少しずつ減らします。`::irmpools` および `::irmreqs` マクロを使用して、接続されているすべてのドライバに対する調整の影響を評価します。

たとえば、ゲストドメインで Oracle Solaris OS のブート中に次の警告が出されるとします。

```
WARNING: emlxs3: interrupt pool too full.
WARNING: ddi_intr_alloc: cannot fit into interrupt pool
```

::irmppools および ::irmreqs マクロは次の情報を表示します。

```
# echo "::irmppools" | mdb -k
ADDR          OWNER   TYPE   SIZE  REQUESTED  RESERVED
00000400016be970 px#0    MSI-X  36    36         36

# echo "00000400016be970::irmreqs" | mdb -k
ADDR          OWNER   TYPE   CALLBACK NINTRS  NREQ  NAVAIL
00001000143acaa8 emlxs#0 MSI-X  No        32      8     8
00001000170199f8 emlxs#1 MSI-X  No        32      8     8
000010001400ca28 emlxs#2 MSI-X  No        32      8     8
0000100016151328 igb#3   MSI-X  No        10      3     3
0000100019549d30 igb#2   MSI-X  No        10      3     3
0000040000e0f878 igb#1   MSI-X  No        10      3     3
000010001955a5c8 igb#0   MSI-X  No        10      3     3
```

この例に示されるデフォルトの制限は 1 デバイスにつき 8 回の割り込みですが、これはシステムへの最後の emlxs3 デバイスの接続を受け入れるのに十分な割り込み数ではありません。emlxs のすべてのインスタンスが同じように動作すると考えれば、emlxs3 はおそらく 8 回の割り込みをリクエストしています。

36 回の割り込みの総プールサイズから、すべての igb デバイスで使用された 12 回の割り込みを引くことにより、24 回の割り込みを emlxs デバイスに使用できます。24 回の割り込みを 4 で割ると、1 デバイスにつき 6 回の割り込みによって、すべての emlxs デバイスが同じパフォーマンスで接続できるようになります。そのため、次の調整が /etc/system ファイルに追加されます。

```
set ddi_msix_alloc_limit = 6
```

/etc/system プロパティ値の正しい作成または更新については、[400 ページの「/etc/system ファイルのプロパティ値の更新」](#)を参照してください。

システムが警告なしで正常にブートすると、::irmppools および ::irmreqs マクロは次の更新された情報を表示します。

```
primary# echo "::irmppools" | mdb -k
ADDR          OWNER   TYPE   SIZE  REQUESTED  RESERVED
00000400018ca868 px#0    MSI-X  36    36         36

# echo "00000400018ca868::irmreqs" | mdb -k
ADDR          OWNER   TYPE   CALLBACK NINTRS  NREQ  NAVAIL
0000100016143218 emlxs#0 MSI-X  No        32      8     6
0000100014269920 emlxs#1 MSI-X  No        32      8     6
```

```

000010001540be30 emlxs#2 MSI-X No      32      8      6
00001000140cbe10 emlxs#3 MSI-X No      32      8      6
00001000141210c0 igb#3  MSI-X No      10      3      3
0000100017549d38 igb#2  MSI-X No      10      3      3
0000040001ceac40 igb#1  MSI-X No      10      3      3
000010001acc3480 igb#0  MSI-X No      10      3      3

```

ドメイン I/O の依存関係の一覧表示

ドメインの I/O 操作は、多くの場合、サービสดメインまたは I/O ドメインなどの別のドメインによって提供されます。たとえば、サービสดメインが仮想デバイスをエクスポートできたり、ルートドメインが物理デバイスへの直接アクセスを提供できたりします。

サービสดメインまたはルートドメインが停止すると依存関係にあるドメインもサービスが中断されるので、これらの暗黙的な I/O の依存関係に注意してください。

`ldm list-dependencies` コマンドを使用すると、ドメイン間の I/O の依存関係を表示できます。ドメインの依存関係の一覧表示に加えて、特定のドメインの依存関係を表示するように出力を反転させることができます。

次の一覧は、`ldm list-dependencies` コマンドを使用して表示できる I/O の依存関係の種類を示しています。

VDISK	仮想ディスクサーバーによってエクスポートされている仮想ディスクバックエンドに仮想ディスクが接続されるときに作成される依存関係
VNET	仮想ネットワークデバイスが仮想スイッチに接続されるときに作成される依存関係
IOV	SR-IOV 仮想機能が SR-IOV 物理機能と関連付けられるときに作成される依存関係

次の `ldm list-dependencies` コマンドは、ドメイン依存関係情報を表示できる方法のいくつかを示しています。

- 詳細なドメイン依存関係情報を表示するには、`-l` オプションを使用します。

```

primary# ldm list-dependencies -l
DOMAIN          DEPENDENCY      TYPE      DEVICE
primary
svcdom
ldg0             primary         VDISK     primary-vds0/vdisk0

```


ldg1	primary	VNET	primary-vsw0/vnet0
		VDISK	svcdom-vds0/vdisk1
		VNET	svcdom-vsw0/vnet1
		VDISK	primary-vds0/vdisk0
	svcdom	VNET	primary-vsw0/vnet0
		IOV	/SYS/MB/NET0/IOVNET.PF0.VF0
		VDISK	svcdom-vds0/vdisk1
		VNET	svcdom-vsw0/vnet1
		IOV	/SYS/MB/NET2/IOVNET.PF0.VF0

- 依存項目の詳細情報を依存関係によってグループ化して表示するには、`-l` オプションと `-r` オプションの両方を使用します。

```
primary# ldm list-dependencies -r -l
```

DOMAIN	DEPENDENT	TYPE	DEVICE
primary	ldg0	VDISK	primary-vds0/vdisk0
		VNET	primary-vsw0/vnet0
	ldg1	VDISK	primary-vds0/vdisk0
		VNET	primary-vsw0/vnet0
		IOV	/SYS/MB/NET0/IOVNET.PF0.VF0
svcdom	ldg0	VDISK	svcdom-vds0/vdisk1
		VNET	svcdom-vsw0/vnet1
	ldg1	VDISK	svcdom-vds0/vdisk1
		VNET	svcdom-vsw0/vnet1
		IOV	/SYS/MB/NET2/IOVNET.PF0.VF0

Logical Domains Manager デーモンの有効化

Logical Domains Manager デーモン `ldmd` は、Oracle VM Server for SPARC ソフトウェアパッケージがインストールされると自動的に有効になります。このデーモンが有効になると、論理ドメインを作成、変更、および制御できます。

SPARC T7 シリーズサーバーおよび SPARC M7 シリーズサーバーでは、ILOM 相互接続サービスによって、`ldmd` デーモンとサービスプロセッサ (SP) の間の通信が可能になります。`ilomconfig-interconnect` サービスはデフォルトで有効です。ILOM 相互接続サービスが有効になっていることを確認するには、[49 ページの「ILOM 相互接続構成を確認する方法」](#)を参照してください。



注意 - `ilomconfig-interconnect` サービスを無効にしないでください。このサービスを無効にすると、論理ドメインや OS が正しく動作しなくなることがあります。

▼ Logical Domains Manager デーモンを有効にする方法

`ldmd` デーモンが無効になっている場合、次の手順に従ってこのデーモンを有効にします。

1. `svcadm` コマンドを使用して、Logical Domains Manager デーモンの `ldmd` を有効にします。

```
# svcadm enable ldmd
```

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

2. Logical Domains Manager が実行中であることを確認します。

`ldm list` コマンドを実行すると、システム上で現在定義されているすべてのドメインが一覧表示されます。特に、`primary` ドメインが表示され、状態が `active` になっているはずです。次のサンプル出力は、システム上に `primary` ドメインのみが定義されていることを示します。

```
# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active ---c-  SP    64    3264M   0.3%  19d 9m
```

自動保存構成データの保存および復元

次のセクションでは、自動保存構成ディレクトリと制約データベースファイルを保存および復元する方法について説明します。

自動保存構成ディレクトリの保存および復元

制御ドメインでオペレーティングシステムを再インストールする前に、自動保存構成ディレクトリを保存し復元することができます。制御ドメイン上でオペレーティングシステムを再インストールした場合は常に、`/var/share/ldomsmanager/autosave-autosave-name` ディレクトリにあるそのドメインの自動保存構成データを保存および復元する必要があります。

`tar` または `cpio` コマンドを使用して、ディレクトリのすべての内容を保存および復元できます。

注記 - 各自動保存ディレクトリには、関連する構成の前の SP 構成更新のタイムスタンプが含まれています。自動保存ファイルを復元すると、タイムスタンプが同期しなくなることがあります。この場合、復元された自動保存構成は、以前の状態 ([newer] または最新) で表示されます。

自動保存構成の詳細については、[ドメイン構成の管理](#)を参照してください。

▼ 自動保存ディレクトリの保存および復元方法

1. 自動保存ディレクトリを保存します。

```
# cd /  
# tar -cvpf autosave.tar var/share/ldomsmanager/autosave-*
```

2. (オプション) クリーンな復元操作を行えるように、既存の自動保存ディレクトリを削除します。

自動保存ディレクトリには、以前の構成によって残されたファイルなどの不要なファイルが含まれていることがあります。このようなファイルは、SP にダウンロードされた構成を破壊することがあります。このような場合、この例に示すとおり、復元操作の前に自動保存ディレクトリを削除します。

```
# cd /  
# rm -rf var/share/ldomsmanager/autosave-*
```

3. 自動保存ディレクトリを復元します。

これらのコマンドは、/var/share/ldomsmanager ディレクトリ内のファイルおよびディレクトリを復元します。

```
# cd /  
# tar -xvpf autosave.tar
```

Logical Domains の制約データベースファイルの保存および復元

制御ドメイン上でオペレーティングシステムをアップグレードした場合は常に、Logical Domains の制約データベースファイル /var/share/ldomsmanager/ldom-db.xml を保存および復元する必要があります。

また、制御ドメインのファイルデータを破損するその他のいずれかの操作 (ディスクスワップなど) を実行した場合は、/var/share/ldomsmanager/ldom-db.xml ファイルも保存および復元します。

注記 - /var/share/ldomsmanager ディレクトリは、すべてのブート環境間で共有されています。/var/opt/SUNWldm は、下位互換性を目的とした /var/share/ldomsmanager へのシンボリックリンクです。

出荷時デフォルト構成とドメインの無効化

初期構成では、プラットフォームが 1 つのオペレーティングシステムのみをホストする単一のシステムとして表示されますが、この構成は出荷時デフォルト構成と呼ばれます。論理ドメインを無効にする場合には、ほかのドメインに割り当てられている可能性のあるすべてのリソース (CPU、メモリー、I/O) にシステムがふたたびアクセスできるように、この構成の復元も必要になる場合があります。

このセクションでは、すべてのゲストドメインを削除し、ドメインのすべての構成を削除し、構成を出荷時のデフォルトに戻す方法について説明します。

▼ すべてのゲストドメインを削除する方法

1. すべてのドメインを停止します。

```
primary# ldm stop-domain -a
```

2. **primary** ドメインを除き、すべてのドメインのバインドを解除します。

```
primary# ldm unbind-domain ldom
```

注記 - 制御ドメインが必要とするサービスを I/O ドメインが提供している場合、その I/O ドメインのバインドを解除できないことがあります。この場合は、この手順をスキップします。

3. **primary** ドメインを除き、すべてのドメインを削除します。

```
primary# ldm remove-domain -a
```

▼ すべてのドメイン構成を削除する方法

1. サービスプロセッサ (SP) 上に格納されているすべてのドメインの構成を一覧表示します。

```
primary# ldm list-config
```

2. **factory-default** 構成を除き、SP に保存されたすべての構成(*config-name*) を削除します。
各構成に対して次のコマンドを使用します。

```
primary# ldm rm-config config-name
```

以前 SP に保存した構成がすべて削除されると、制御ドメイン (primary) のリブート時に次に使用されるドメインが **factory-default** になります。

▼ 出荷時デフォルト構成を復元する方法

1. 出荷時デフォルト構成を選択します。

```
primary# ldm set-config factory-default
```

2. 制御ドメインを停止します。

```
primary# shutdown -i5 -g0 -y
```

3. システムの電源再投入を実行して、出荷時デフォルト構成をロードします。

```
-> stop /SYS  
-> start /SYS
```

▼ Logical Domains Manager を無効にする方法

Logical Domains Manager を無効にしても動作中のドメインは停止しませんが、新しいドメインの作成、既存のドメインの構成の変更、またはドメインの状態のモニターを行う機能は無効になります。



注意 - Logical Domains Manager を無効にすると、エラー報告や電源管理などの一部のサービスが無効になります。エラー報告については、**factory-default** 構成の場合は、制御ドメインをリブートしてエラーの報告を復元することはできます。ただし、電源管理を再度有効にすることはできません。また、一部のシステム管理またはモニタリングツールは Logical Domains Manager に依存しています。

- 制御ドメインから Logical Domains Manager を無効にします。

```
primary# svcadm disable ldmd
```

▼ サービスプロセッサから出荷時デフォルト構成を復元する方法

サービスプロセッサから出荷時デフォルト構成を復元できます。

1. サービスプロセッサから出荷時デフォルト構成を復元します。

```
-> set /HOST/bootmode config=factory-default
```
2. システムの電源再投入を実行して、出荷時デフォルト構成をロードします。

```
-> reset /SYS
```

パート II

オプションの Oracle VM Server for SPARC ソフトウェア

ここでは、Oracle VM Server for SPARC 3.3 ソフトウェアに使用できるオプションのソフトウェアと機能について説明します。

- [第18章「Oracle VM Server for SPARC テンプレートの使用」](#)
- [第19章「Oracle VM Server for SPARC 物理から仮想への変換ツール」](#)
- [第20章「電源管理の使用」](#)
- [第21章「Oracle VM Server for SPARC 管理情報ベース \(MIB\) ソフトウェアの使用」](#)
- [第22章「Logical Domains Manager の検出」](#)
- [第23章「Logical Domains Manager での XML インタフェースの使用」](#)

◆◆◆ 第 18 章 18

Oracle VM Server for SPARC テンプレートの使用

この章では、次の項目について説明します。

- [433 ページの「Oracle VM Server for SPARC テンプレートの概要」](#)
- [434 ページの「Oracle VM Server for SPARC テンプレートユーティリティのインストール」](#)
- [434 ページの「Oracle VM Server for SPARC テンプレートのライフサイクル」](#)
- [438 ページの「Oracle VM Server for SPARC テンプレートの例」](#)

Oracle VM Server for SPARC テンプレートの概要

Oracle VM Server for SPARC テンプレートコマンドを使用すると、SPARC システム用の Oracle VM Server for SPARC テンプレートを作成、配備、および構成できます。これらのコマンドは、ディスクイメージファイル、およびアーカイブ (.ova) に含まれているプロパティの XML 記述子を含む OVF テンプレート仕様に基づいています。

これらのコマンドは、Oracle VM Server for SPARC システムの制御ドメインで実行します。コマンド行での実行に加えて、これらのコマンドは完全に自動化された方法で実行したり、より大きなワークフローの一部としてほかのプログラムから実行したりすることもできます。

Oracle VM Server for SPARC テンプレートコマンドは次のとおりです。

- `ovmtconfig` – ドメインのファイルシステムに対して直接、構成操作を実行します。このコマンドは、ドメインディスクおよび関連付けられた ZFS ファイルシステムをバックマウントしてから、ファイルシステムに対してコマンドを実行します。これらのコマンドは、プロパティファイルによって、およびプロパティの名前と値のペアを指定することによってさらに制御できます。[ovmtconfig\(1M\)](#) のマニュアルページを参照してください。

ovmtconfig コマンドでは、ゲストドメインが、制御ドメインで実行されているバージョンより新しくないバージョンの Oracle Solaris OS を実行している必要があることに注意してください。

- ovmtcreate – 既存の Oracle VM Server for SPARC ドメインからテンプレートを作成します。[ovmtcreate\(1M\)](#) のマニュアルページを参照してください。
- ovmtdeploy – Oracle VM Server for SPARC テンプレートからドメインを作成します。[ovmtdeploy\(1M\)](#) のマニュアルページを参照してください。
- ovmtlibrary – ファイルを構成し、データベース内の情報を格納、取得、および編集することによって、Oracle VM Server for SPARC テンプレートのためのデータベースおよびファイルシステムベースのリポジトリを管理します。[ovmtlibrary\(1M\)](#) のマニュアルページを参照してください。
- ovmtprop – Oracle Solaris OS のプロパティを表示および設定できるようにします。このプロパティは、名前と値のペアとして指定されます。[ovmtprop\(1M\)](#) のマニュアルページを参照してください。

Oracle VM Server for SPARC テンプレートユーティリティのインストール

制御ドメインに Oracle VM Server for SPARC テンプレートユーティリティのソフトウェアパッケージをインストールするには、まず Oracle VM Server for SPARC 3.3 ソフトウェアをダウンロードして制御ドメインにインストールする必要があります。『[Oracle VM Server for SPARC 3.3 インストールガイド](#)』の第 2 章「ソフトウェアのインストール」を参照してください。

次に、Oracle VM Server for SPARC テンプレートユーティリティのソフトウェアパッケージ ovmtutils をインストールします。

```
# pkg install -v -g IPS-package-directory/ldoms.repo ovmtutils
```

Oracle VM Server for SPARC テンプレートのライフサイクル

このセクションでは、テンプレート作成プロセスの各段階、実行されるアクション、および Oracle VM Server for SPARC テンプレートユーティリティを使用してこのプロセスを支援する方法について説明します。

注記 - アプリケーションや初回ブートスクリプトを使用したテンプレートの作成および開発は繰り返しのプロセスです。ソースコード管理システムを使用してスクリプトとプロパティを管理することにより、構成のすべての側面を同期させるように注意してください。

次に、テンプレート作成プロセスの各段階、実行されるアクション、および Oracle VM Server for SPARC テンプレートユーティリティを使用してこのプロセスを支援する方法について説明します。

1. **テンプレートの作成。** 事前に作成された汎用のテンプレートを使用できますが、既存のドメインからカスタムテンプレートを作成できます。このドメインには、必要なすべてのオペレーティングシステムコンポーネント、アプリケーションソフトウェア、およびその他のユーティリティが完全にインストールされている必要があります。

この環境は通常、環境を完成させるために必要なアクションがごく少数になるように、できるだけ完全に構成されます。メモリー、仮想 CPU、仮想ネットワーク、ディスクなどのすべてのドメイン設定に、必要とする配備が反映されているようにしてください。

この段階では、1 つまたは複数の「初回ブート」スクリプトを作成します。これらのスクリプトは、指定したプロパティに基づいて最終的な構成を実行する環境に含めます。これらのプロパティは必ず、テンプレートごとの README ファイルに記録し、説明を加えるようにしてください。

注記 - いずれかの初回ブートスクリプトがドメイン変数にアクセスする場合は、ゲストドメインに `ovmtprop` ユーティリティがインストールされていることを確認してください。

2. **テンプレートの作成。** テンプレートを作成する前に、ソースドメインの環境が確実に未構成の状態であるようにして、規定されたアクション (初回ブートスクリプトの一部である場合が多い) によってあとで構成できるようにします。

たとえば、次の手順を実行します。

- アプリケーション固有の構成をすべて削除し、あとで再作成されるようにします。
- 構成ファイルのデフォルト値を使用します。
- システム名、ネットワーク構成、パスワードなどの Oracle Solaris OS 構成情報をすべてリセットするようにしてください。これらの構成情報は、プロパティ値や構成スクリプトによってあとで指定されます。
- ルートファイルシステム以外の `zpool` をすべてエクスポートして、新しいドメインで認識できるようにします。

これらの手順を実行したら、ドメインを停止し、`ovmtcreate` ユーティリティーを実行して、そのドメインからテンプレートを作成できます。

3. **テンプレートの名前の指定。**次の形式を使用します。

`technology.OS.application.architecture.build.ova`

たとえば、次のテンプレート名は、SPARC プラットフォーム上で Oracle Solaris 11.2 OS のビルド 2 を実行し、WebLogic サーバーのバージョン 12.1.2 を実行するドメインに対するものです: `OVM_S11.2_WLS12.1.2_SPARC_B2.ova`

4. **テンプレートの配布。**このテンプレートは、`.ova` の拡張子を持つ 1 つのファイルです。このファイルには、配備に必要な圧縮されたディスクイメージとメタデータが含まれています。このテンプレートにはまた、ペイロードファイルチェックサムのマニフェストファイルも含まれています。これは、配布以降に内容が変更されていないことを検証するための全体的なアーカイブチェックサムと結合されている場合があります。

Web ベースのサービスを使用してテンプレートを配布することも、テンプレートを複製する代わりに中央リポジトリを保持することもできます。

5. **テンプレートの配備。**テンプレートにはソースドメインによって認識されるシステムの側面のみが取得されるため、テンプレートの配備をサポートするにはどのサービスが存在している必要があるかを理解する必要があります。

必要なサービスには次のものが含まれます。

- テンプレートからの仮想ネットワークが接続される可能性のある適切なインタフェースへの 1 つまたは複数の仮想スイッチ
- 仮想ディスクサービス
- コンソールサービス
- テンプレートの要件に対応できる十分な仮想 CPU およびメモリー

`ovmtdeploy` ユーティリティーはこれらの設定の多くをオーバーライドできますが、テンプレートで提供される最小値がベースライン要件を表します。

`ovmtdeploy` ユーティリティーを使用すると、仮想ディスクを自動的に抽出し、圧縮解除して配備ディレクトリにコピーしたり、テンプレートに記述されているさまざまな仮想デバイスを構築したりできます。

この時点でドメインを起動できますが、ドメインが完全に機能するには、ドメインコンソールを使用して手動の構成手順をいくつか実行することが必要になる可能性があります。

6. **ドメインの自動的な構成。**テンプレートによって作成されたドメインの構成は、いくつかのタイプのアクションで構成されます。たとえば、初回ブートスクリプトに構成対象の情報を提供

するために、プロパティの名前と値のペアを指定することがあります。また、ドメインのファイルシステムに対して構成ファイルのコピーなどのアクションを実行するために、仮想ディスクを制御ドメインにバックマウントすることもあります。

`ovmtconfig` ユーティリティはこれらのドメイン構成アクティビティを自動化し、1 つまたは複数のコマンドスクリプトやプロパティファイルを指定することにより、実行するアクションやドメインを構成するために使用するプロパティを指定できるようにします。

Oracle Solaris OS を構成するために、`ovmtconfig` ユーティリティはドメインのルートファイルシステムをバックマウントし、指定された構成スクリプトやプロパティから `sc_profile.xml` ファイルを作成します。このプロファイルにより、Oracle Solaris OS は最初のブート時に自身を構成できます。

7. **最初の構成。** Oracle Solaris OS が正常に構成され、最初のブートが終了したら、インストールされているアプリケーションをすべて構成する必要があります。構成フェーズ中に、`ovmtconfig` ユーティリティは、次のいずれかの方法を使用して配備対象のドメインに構成情報を渡します。

- **直接のアクション** – `ovmtconfig` ユーティリティは、ゲストドメインのファイルシステムを制御ドメインにバックマウントし、ファイルおよびファイルシステムに対して直接のアクションを実行します。これらのアクションには、構成ファイルの作成やシステムバイナリのコピーが含まれることがあります。これらのアクションは、`ovmtconfig` ユーティリティに指定したスクリプトに記述されています。

ゲストドメイン内で実行されるように設計されているプロセスは、制御ドメインに影響を与える可能性があるため、通常これらのアクションには含まれません。実行するコマンドを指定するには、`ovmtconfig -c` コマンドを使用します。

- **ドメイン変数** – ローカルプロパティファイルに加えて、制御ドメインで `ovmtconfig` ユーティリティを実行することにより、あとでゲストドメインで `ovmtprop` ユーティリティが使用できるドメイン変数を設定できます。この方法を使用すると、構成が完了したあとに初回ブートスクリプトがこれらのプロパティに直接アクセスし、ゲストドメインに直接構成情報を提供できます。



注意 - ドメインにパスワードなどの機密情報を渡すために、暗号化されていないプロパティを使用しないでください。Oracle Solaris OS を構成するために使用されるもの以外のプロパティは、ldm 変数として平文でドメインに渡されます。これらのプロパティ値は、ldm コマンドの実行を承認されている制御ドメイン上のユーザー、および配備されたドメインにログインしているユーザーに表示されます。

この問題を回避するには、ゲストドメインを配備し、アプリケーションが完全に構成されたあとにドメイン変数を手動で削除します。

ドメイン変数の前には `OVMTVAR_ID` の文字列が付加されます。ここで、*ID* は 4 桁です。次の例は、システムのネットマスクに対するドメイン変数宣言を示しています: `OVMTVAR_1006=ovmt.prop.key.com.oracle.solaris.network.netmask.0=24`。

制御ドメインから、`ldm list-variable | grep OVMTVAR_` コマンドを使用してドメイン変数を一覧表示します。次に、`ldm remove-variable OVMTVAR_ID` コマンドを使用して、機密情報を含む変数をすべて削除します。

たとえば、ゲストで `ovmtprop` を実行するスーパーバイザースクリプトを使用し、制御ドメインから `ovmtconfig -v` を実行することによって、ネットワークにアクセスできない構成の側面の変更を自動化できます。

この時点で、このドメインは完全に構成され、動作可能になります。

Oracle VM Server for SPARC テンプレートの例

このセクションでは、次の Oracle VM Server for SPARC テンプレートタスクの例を示します。

- [例18-1「Oracle VM Server for SPARC テンプレートの作成」](#)
- [例18-2「Oracle VM Server for SPARC テンプレートのプロパティの構成」](#)
- [例18-3「Oracle VM Server for SPARC テンプレートの配備」](#)
- [例18-4「Oracle VM Server for SPARC テンプレートライブラリの管理」](#)

例 18-1 Oracle VM Server for SPARC テンプレートの作成

次の `ovmtcreate` コマンドは、`ovmtcreate_example` という名前の `ldg1` ドメインに基づいたテンプレートを作成します。結果として得られるテンプレート名には `.ova` の接尾辞が付くことに注意してください。

```
# ovmtcreate -d ldg1 -o ovmtcreate_example
Oracle Virtual Machine for SPARC Template Creation Utility
ovmtcreate Version: 3.3.0.0.12
Copyright (c) 2014, 2015, Oracle and/or its affiliates. All rights reserved.

STAGE 1 - EXAMINING SYSTEM AND ENVIRONMENT
-----
Performing platform & prerequisite checks
Checking user permissions
Checking for required packages
Checking for required services
Checking directory permissions

STAGE 2 - ANALYZING DOMAIN
-----
Retrieving and processing attributes
Checking domain state
Getting domain resource settings
Discovering network topology
Discovering disk topology

STAGE 3 - ARCHIVE CREATION
-----
Checking destination and current directory capacity
Compressing disk image
Creating XML configuration
Calculating manifest checksums
Creating archive file
Checking archive

PROCESS COMPLETED
-----
Started: Tue Aug 18 15:29:14 PDT 2015
Completed: Tue Aug 18 15:41:25 PDT 2015
Elapsed time: 0:12:11
```

例 18-2 Oracle VM Server for SPARC テンプレートのプロパティの構成

`ovmtconfig` および `ovmtprop` ユーティリティを使用すると、それぞれ Oracle VM Server for SPARC テンプレートのプロパティ値と Oracle Solaris OS のプロパティ値を指定できます。

- 次の `ovmtconfig` コマンドは、`ldg1` ドメインのファイルシステムに対して直接、構成操作を実行します。

-c オプションは、プロパティー値を設定するための /opt/ovmtutils/share/scripts/ovmt_s11_scprofile.sh スクリプトを指定します。-p オプションは、com.oracle.solaris.network.ipaddr および com.oracle.solaris.system.computer-name プロパティーの特定の値を指定します。

```
# ovmtconfig -v -d ldg1 -f -s \
-c /opt/ovmtutils/share/scripts/ovmt_s11_scprofile.sh \
-p com.oracle.solaris.network.ipaddr.0=10.153.118.211,\
com.oracle.solaris.system.computer-name=system1
```

Oracle Virtual Machine for SPARC Configuration Utility

ovmtconfig Version: 3.3.0.0.12

Copyright (c) 2014, 2015, Oracle and/or its affiliates. All rights reserved.

STAGE 1/7 - EXAMINING SYSTEM AND ENVIRONMENT

Checking operating system

Checking platform type

Checking user permissions

Checking packages

Checking host domain name

Checking host domain type

Checking services

STAGE 2/7 - PROCESSING COMMAND LINE OPTIONS

Parsing individual properties

Creating consolidated properties list

STAGE 3/7 - ANALYZING TARGET DOMAIN

Stopping domain ldg1

Analyzing domain disk topology for domain ldg1

Discovering 1 volumes for vDisks

Examining 1 backend devices

unbinding domain ldg1

Creating 1 virtual disks for back mount

Created virtual disk 0

STAGE 4/7 - PERFORMING BACKMOUNT

Finding Solaris device for vdisks

Importing zpools for 1 Solaris devices

Detected conflicting zpool name, attempting rename

Getting boot file system for properties in 1 zpool

Setting properties in 1 zpools

Mounting ZFS file systems

Mounting ZFS found in zpool rpool_1

STAGE 5/7 - PERFORMING ACTIONS ON TARGET DOMAIN

STAGE 6/7 - UNMOUNTING AND RESTORING DOMAIN STATE

Rolling back commands DEBUG [20150819-07:02:42]: Rolling back 8 /usr/sbin/zfs unmount -f
rpool_1/ROOT/solaris/var
completed

STAGE 7/7 - SETTING TARGET DOMAIN ENVIRONMENT

Checking 2 properties to set as domain variables

/opt/ovmtutils/bin/agent/lib/ldoms/ldmxml.py:1692: FutureWarning: The behavior of this
method will change in future versions. Use specific
'len(elem)' or 'elem is not None' test instead.

if outgoing:

.. Starting domain

Process completed

- 次の ovmtprop コマンドは、Oracle Solaris OS のプロパティ値を指定します。

```
# ovmtprop set-prop com.oracle.solaris.system.computer-name=test ldg1
```

Oracle Virtual Machine for SPARC Properties Utility

ovmtprop Version: 3.3.0.0.12

Copyright (c) 2014, 2015, Oracle and/or its affiliates. All rights reserved.

```
/opt/ovmtutils/bin/agent/lib/ldoms/ldmxml.py:1692: FutureWarning: The behavior of this
method will change in future versions. Use
specific 'len(elem)' or 'elem is not None' test instead.
```

```
if outgoing:
```

```
solaris property set via VM-API
```

com.oracle.solaris.system.computer-name プロパティの値が test であることを確認するには、ldm list -l コマンドを使用します。

```
# ldm ls -l ldg1
NAME STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
ldg1 active -n---- 5000 16    16G    0.0% 0.0% 2h 7m
..
CONSTRAINT

VARIABLES
auto-boot?=true

VMAPI TO GUEST
com.oracle.solaris.fmri.count=0
com.oracle.solaris.system.computer-name=test
```

例 18-3 Oracle VM Server for SPARC テンプレートの配備

次の ovmtdeploy コマンドは、/export/ovmtdeploy ディレクトリ内の ovmtcreate_example.ova Oracle VM Server for SPARC テンプレートを使用して、ldg1 という名前のドメインを作成します。

```
# ovmtdeploy -d ldg1 -o /export/ovmtdeploy ovmtcreate_example.ova
Oracle Virtual Machine for SPARC Deployment Utility
ovmtdeploy Version 3.3.0.0.12
Copyright (c) 2014, 2015, Oracle and/or its affiliates. All rights reserved.

STAGE 1 - EXAMINING SYSTEM AND ENVIRONMENT
-----
Checking user privilege
Performing platform & prerequisite checks
Checking for required services
Named resourced available

STAGE 2 - ANALYZING ARCHIVE & RESOURCE REQUIREMENTS
-----
```

```

Checking .ova format and contents
Validating archive configuration
Checking sufficient resources present
WARNING: Virtual switch primary-vsw0 already exists

```

STAGE 3 - EXTRACTING ARCHIVE

```

-----
Extracting archive
Validating checksums
Decompressing disk image(s)

```

STAGE 4 - DEPLOYING DOMAIN

```

-----
Creating domain and adding resources
Validating deployment
Domain created:

```

ldm list の出力は、ldg1 という名前の新しいドメインが作成されたことを示しています。

```

# ldm list
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary   active -n-cv-  UART  8      40G     1.4%  1.1%  6d 2h 18m
ldg1      active -n----  5000  8      8G      41%   38%  28s

```

例 18-4 Oracle VM Server for SPARC テンプレートライブラリの管理

ovmtlibrary コマンドは、ファイルを構成し、データベース内の情報を格納、取得、および編集することによって、Oracle VM Server for SPARC テンプレートのためのデータベースおよびファイルシステムベースのリポジトリを管理します。

- 次のコマンドは、export/user1/ovmtlibrary_example 内にテンプレートライブラリを作成します。

```
# ovmtlibrary -c init -l /export/user1/ovmtlibrary_example
```

```

Oracle Virtual Machine for SPARC Template Library Utility
ovmtlibrary Version: 3.3.0.0.12
Copyright (c) 2015 Oracle and/or its affiliates. All rights reserved.
Init complete

```

- 次のコマンドは、sol-11_2-ovm-2-sparc.ova テンプレートを export/user1/ovmtlibrary_example ライブラリ内に格納します。

```
# ovmtlibrary -c store -d "ovmtlibrary example" -o http://system1.example.com/s11.2/
templates/sol-11_2-ovm-2-sparc.ova -l /export/user1/ovmtlibrary_example
```

```

Oracle Virtual Machine for SPARC Template Library Utility

```

```

ovmtlibrary Version: 3.3.0.0.12
Copyright (c) 2015 Oracle and/or its affiliates. All rights reserved.

Templates present in path "/export/user1/ovmtlibrary_example"

event id is 2

*****
converted 'http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-sparc.ova' (646) ->
'http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-sparc.ova' (UTF-8)
--2015-08-18 16:37:17-- http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-
sparc.ova
Resolving system1.example.com (system1.example.com)... 10.134.127.18
Connecting to system1.example.com (system1.example.com)|10.134.127.18|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1018341888 (971M) [text/plain]
Saving to: '/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-2-
sparc.ova'

/export/user1/ovmtlibrary_example/repo 100%
[=====] 971.17M
6.05MB/s in 5m 37s
2015-08-18 16:42:55 (2.88 MB/s) - '/export/user1/ovmtlibrary_example/repository/
templates/1/1/sol-11_2-ovm-2-sparc.ova' saved
[1018341888/1018341888]
*****
Download complete
Extracting the ova file...
Extract complete
Decompress file System.img.gz
Store complete
■ 次のコマンドは、export/user1/ovmtlibrary_example ライブラリの内容を一覧表示しま
す。

# ovmtlibrary -c list -l /export/user1/ovmtlibrary_example
Oracle Virtual Machine for SPARC Template Library Utility
ovmtlibrary Version: 3.3.0.0.12

```

Copyright (c) 2015 Oracle and/or its affiliates. All rights reserved.

Templates present in path "/export/user1/ovmtlibrary_example"

ID	Name	Version	Description	Date
1	sol-11_2-ovm-2-sparc	1	ovmtlibrary example	2015-08-18

- 次のコマンドは、export/user1/ovmtlibrary_example ライブラリの詳細な一覧を表示します。

```
# ovmtlibrary -c list -i 1 -o -l /export/user1/ovmtlibrary_example
```

Oracle Virtual Machine for SPARC Template Library Utility

ovmtlibrary Version: 3.3.0.0.12

Copyright (c) 2015 Oracle and/or its affiliates. All rights reserved.

Templates present in path "/export/user1/ovmtlibrary_example"

ID	Name	Type	Path	Size(bytes)
1	sol-11_2-ovm-2-sparc.ova	ova	/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-2-sparc.ova	1018341888
2	sol-11_2-ovm-sparc.ovf	ovf	/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-sparc.ovf	3532
3	sol-11_2-ovm-sparc.mf	mf	/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-sparc.mf	137
4	System.img	img	/export/user1/ovmtlibrary_example/repository/templates/1/1/System.img	21474836480

◆◆◆ 第 19 章 19

Oracle VM Server for SPARC 物理から仮想への変換ツール

この章では、次の項目について説明します。

- 447 ページの「Oracle VM Server for SPARC P2V ツールの概要」
- 450 ページの「バックエンドデバイス」
- 451 ページの「Oracle VM Server for SPARC P2V ツールのインストールおよび構成」
- 454 ページの「ldmp2v コマンドの使用」
- 463 ページの「Oracle VM Server for SPARC 物理から仮想への変換ツールの既知の問題」

注記 - ldmp2v ユーティリティは、バグ修正や改善要求に対応するように更新されなくなります。このユーティリティはサポートされなくなりますが、引き続き Oracle VM Server for SPARC ソフトウェアの一部として組み込まれて文書化されます。

Oracle VM Server for SPARC P2V ツールの概要

Oracle VM Server for SPARC Physical-to-Virtual (P2V) 変換ツールは、既存の物理システムを、チップマルチスレッディング (CMT) システム上の論理ドメインで Oracle Solaris 10 OS を実行する仮想システムに自動的に変換します。Oracle Solaris 11 OS を実行している制御ドメインから ldmp2v コマンドを実行して、次のソースシステムのいずれかを論理ドメインに変換できます。

- Solaris 8、Solaris 9、または Oracle Solaris 10 OS を実行している、任意の sun4u SPARC ベースのシステム

- Oracle Solaris 10 OS を実行しているが、論理ドメインでは実行していない sun4v システム

注記 - `ldmp2v` コマンドは、ZFS ルートがある Oracle Solaris 10 OS または Oracle Solaris 11 OS を実行する SPARC ベースのシステムをサポートしません。

物理システムから仮想システムへの変換は、次のフェーズで実行されます。

- **収集フェーズ**。物理ソースシステムで実行します。収集フェーズでは、ソースシステムについて収集された構成情報に基づいて、ソースシステムのファイルシステムイメージが作成されます。
- **準備フェーズ**。ターゲットシステムの制御ドメイン上で実行されます。準備フェーズでは、収集フェーズで収集された構成情報に基づいて、ターゲットシステムに論理ドメインが作成されます。ファイルシステムイメージは、1 つ以上の仮想ディスクに復元されます。P2V ツールを使用して、プレーンファイルまたは ZFS ボリュームに仮想ディスクを作成できます。また、物理ディスクや LUN、または作成したボリュームマネージャーのボリュームに仮想ディスクを作成することもできます。このイメージは、論理ドメインとして動作できるように変更されます。
- **変換フェーズ**。ターゲットシステムの制御ドメイン上で実行されます。`convert` フェーズで、標準の Oracle Solaris アップグレードプロセスを使用することにより、作成された論理ドメインが、Oracle Solaris 10 OS を実行する論理ドメインに変換されます。

P2V ツールについては、[ldmp2v\(1M\)](#) のマニュアルページを参照してください。

次のセクションからは、物理システムから仮想システムへの変換が実行される方法について説明します。

収集フェーズ

収集フェーズは、変換対象のシステムで実行されます。一貫性のあるファイルシステムイメージを作成するには、システムの動作を最小限に抑えて、すべてのアプリケーションを停止する必要があります。論理ドメインに移動されるすべてのファイルシステムが確実にマウントされるようにするため、`ldmp2v` コマンドにより、すべてのマウント済み UFS ファイルシステムのバックアップが作成されます。移動する必要がないマウント済みファイルシステム (SAN ストレージ上のファイルシステムや、他の手段で移動するファイルシステムなど) は除外することができます。そのようなファイルシステムを除外するには、`-x` オプションを使用します。`-x` オプションにより除外さ

れたファイルシステムは、ゲストドメイン上に再作成されません。`-o` オプションを使用して、ファイルとディレクトリを除外できます。

ソースシステムでの変更は不要です。唯一の条件は、制御ドメイン上に `ldmp2v` スクリプトがインストールされていることです。ソースシステムに `flarcreate` ユーティリティーが存在していることを確認してください。

準備フェーズ

準備フェーズでは、収集フェーズで収集されたデータを使用して、ソースシステムに相当する論理ドメインを作成します。

次のいずれかの方法で `ldmp2v prepare` コマンドを使用できます。

- **自動モード。**このモードは自動的に仮想ディスクを作成し、ファイルシステムデータを復元します。
 - ソースシステム上にあるものと同じサイズで、論理ドメインと必要な仮想ディスクを作成します。
 - ディスクをパーティションに分割し、ファイルシステムを復元します。

、`/usr`、および `/var` ファイルシステムの合計サイズが 10 ギガバイト未満であれば、Oracle Solaris 10 OS の大容量ディスクに関する要件を満たすように、これらのファイルシステムのサイズが自動的に調整されます。`-x no-auto-adjust-fs` オプションを使用するか、`-m` オプションを使用してファイルシステムのサイズを手動で変更することで、自動サイズ変更を無効にできます。
 - 論理ドメインの OS イメージを変更して、物理ハードウェアへのすべての参照を、論理ドメインに適したバージョンに置き換えます。これにより、通常の Oracle Solaris アップグレードプロセスを使用して、システムを Oracle Solaris 10 OS にアップグレードできます。変更には、`/etc/vfstab` ファイルを更新して新しいディスク名を記述することが含まれます。Oracle Solaris Volume Manager または Veritas Volume Manager (VxVM) によりカプセル化されているすべてのブートディスクは、このプロセス中に自動的にカプセル化を解除されます。カプセル化が解除されたディスクは、プレーンディスクスライスに変換されます。VxVM がソースシステムにインストールされている場合、P2V プロセスにより、作成されたゲストドメイン上で VxVM が無効になります。
- **非自動モード。**手動で仮想ディスクを作成し、ファイルシステムデータを復元する必要があります。このモードでは、ディスクのサイズと数、パーティション分割、およびファイルシステムの

配置を変更できます。このモードの準備フェーズでは、論理ドメインの作成と、OS イメージの変更手順のみをファイルシステム上で実行します。

- **クリーンアップモード。**ldmp2v によって作成された論理ドメインとその下のすべてのバックエンドデバイスを削除します。

変換フェーズ

変換フェーズでは、論理ドメインは Oracle Solaris アップグレードプロセスを使用して Oracle Solaris 10 OS にアップグレードされます。アップグレード処理は、既存のすべてのパッケージを削除し、Oracle Solaris 10 sun4v パッケージをインストールします。これにより、sun4u から sun4v への変換は自動的に実行されます。convert フェーズでは、Oracle Solaris DVD ISO イメージ、またはネットワークインストールイメージを使用できます。Oracle Solaris 10 システムでは、Oracle Solaris JumpStart 機能を使用して、完全に自動化されたアップグレード処理を実行することもできます。

バックエンドデバイス

ゲストドメイン用の仮想ディスクは、多くのバックエンドの種類 (ファイル (file)、ZFS ボリューム (zvol)、物理ディスクまたは LUN (disk)、またはボリュームマネージャーのボリューム (disk)) について作成できます。ldmp2v コマンドは、次のいずれかの方法でバックエンドの種類として file または zvol が指定された場合、適切なサイズのファイルまたは ZFS ボリュームを自動的に作成します。

- **-b オプションを使用する**
- **/etc/ldmp2v.conf ファイルの BACKEND_TYPE パラメータの値で指定する**

バックエンドの種類が disk の場合、仮想ディスクのバックエンドデバイスとして物理ディスク、LUN、またはボリュームマネージャーのボリューム (Oracle Solaris Volume Manager および Veritas Volume Manager (VxVM)) を使用できます。準備フェーズの開始前に、適切なサイズのディスクまたはボリュームを作成する必要があります。物理ディスクまたは LUN の場合、ディスクのブロックまたは文字デバイスのスライス 2 として、たとえば /dev/dsk/c0t3d0s2 のようにバックエンドデバイスを指定します。ボリュームマネージャーのボリュームの場合、ボリュームのブロックまたは文字デバイスを、たとえば Oracle Solaris Volume Manager では /dev/md/dsk/d100、VxVM では /dev/vx/dsk/ldomdg/vol1 のように指定します。

-B *backend:volume:vdisk* オプションでボリューム名と仮想ディスク名を指定しない限り、ゲスト用に作成するボリュームと仮想ディスクにはデフォルトの名前が付けられます。

- *backend* は、使用するバックエンドの名前を指定します。バックエンドの種類が *disk* の場合は、*backend* を指定する必要があります。バックエンドの種類が *file* および *zvol* の場合は *backend* はオプションで、*ldmp2v* で作成されるファイルまたは ZFS ボリュームにデフォルト以外の名前を設定するために使用できます。デフォルトの名前は `$BACKEND_PREFIX/guest-name/diskN` です。
- すべてのバックエンドの種類について *volume* はオプションで、ゲストドメインに作成する仮想ディスクサーバーボリュームの名前を指定します。指定されない場合、*volume* は *guest-name-volN* です。
- すべてのバックエンドの種類について *vdisk* はオプションで、ゲストドメインのボリューム名を指定します。指定されない場合、*vdisk* は *diskN* です。

注記 - 変換プロセス中、制御ドメイン内で名前を確実に一意にするために、仮想ディスクは一時的に *guest-name-diskN* という名前に変更されます。

backend、*volume*、または *vdisk* に空白の値を指定するには、コロン区切り文字だけを入力します。たとえば、-B *::vdisk001* と指定すると、仮想ディスク名が *vdisk001* に設定され、バックエンドとボリュームにはデフォルトの名前が使用されます。*vdisk* を指定しない場合は、最後のコロン区切り文字を省略できます。たとえば、-B */ldoms/ldom1/vol001:vol001* とした場合、バックエンドファイル名が */ldoms/ldom1/vol001*、ボリューム名が *vol001* に指定されます。デフォルトの仮想ディスク名は *disk0* です。

Oracle VM Server for SPARC P2V ツールのインストールおよび構成

Oracle VM Server for SPARC P2V ツールパッケージは、「ターゲットシステムの制御ドメインにのみ」インストールして構成する必要があります。パッケージをソースシステムにインストールする必要はありません。その代わりに、ターゲットシステムの */usr/sbin/ldmp2v* スクリプトを単にソースシステムにコピーします。

注記 - *ldmp2v* は、*ldomsmanager* パッケージからデフォルトでインストールされます。

SPARC P2V ツールを使用するための前提条件

Oracle VM Server for SPARC P2V ツールを実行する前に、次の条件が満たされていることを確認します。

- ソースシステムに、次の Flash アーカイブパッチがインストールされている。
 - **Solaris 8 OS の場合:** パッチ ID 109318-34 以降
 - **Solaris 9 OS の場合:** パッチ ID 113434-06 以降
- ターゲットシステムの制御ドメインは、少なくとも Oracle Solaris 11 OS を実行します
- ゲストドメインが、Oracle Solaris 10 5/08 OS 以降を実行している
- ソースシステムが、Solaris 8 OS 以降を実行している

これらの必要条件のほかに、NFS ファイルシステムがソースシステムとターゲットシステムの両方で共有されるように構成する必要があります。このファイルシステムは、root が書き込みできるようにしてください。ただし、共有ファイルシステムを使用できない場合は、ソースシステムとターゲットシステムの両方でソースシステムのファイルシステムダンプ出力を格納できる大きさのローカルファイルシステムを使用します。

SPARC P2V ツールを使用する場合の制限事項

Oracle VM Server for SPARC P2V ツールには、次の制限事項があります。

- UFS ファイルシステムのみがサポートされています。
- プレーンディスク (/dev/dsk/c0t0d0s0)、Oracle Solaris Volume Manager のメタデバイス (/dev/md/dsk/dNNN)、および VxVM のカプセル化されたブートディスクのみが、ソースシステムでサポートされます。
- P2V プロセス中に、それぞれのゲストドメインに存在できるのは単一の仮想スイッチと仮想ディスクサーバーのみです。P2V 変換のあとで、仮想スイッチと仮想ディスクサーバーをドメインに追加できます。
- VxVM ボリュームのサポートは、カプセル化されたブートディスクの rootvol、swapvol、usr、var、opt、および home ボリュームに制限されます。これらのボリュームの元のスライスは、ブートディスクに存在している必要があります。P2V ツールは、Oracle Solaris 10 OS 上の Veritas Volume Manager 5.x をサポートします。ただし、P2V ツールを使用して、VxVM を使用する Solaris 8 および Solaris 9 オペレーティングシステムを変換することもできます。

- ゾーンのある Oracle Solaris 10 システムは、`ldmp2v collect` 操作を実行する前に、`zoneadm detach` コマンドを使用して、ゾーンが切り離されている場合に変換できます。P2V 変換が完了したら、`zoneadm attach` コマンドを使用して、ゲストドメインに作成されていたゾーンを再接続します。ゲストドメインでのこれらの手順の実行については、『[System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones](#)』の第 23 章「Moving and Migrating Non-Global Zones (Tasks)」を参照してください。

注記 - P2V ツールは、ゾーンパスやネットワークインタフェースなどのゾーン構成を更新したり、ゾーンパスのストレージを移動または構成したりしません。手動でゾーン構成を更新し、ゲストドメインにゾーンパスを移動する必要があります。『[System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones](#)』の第 23 章「Moving and Migrating Non-Global Zones (Tasks)」を参照してください。

▼ Oracle VM Server for SPARC P2V ツールを構成する方法

`ldmp2v` コマンドは、`ldomsmanager` パッケージをインストールするとデフォルトでインストールされます。

1. 管理者になります。

Oracle Solaris 11.3 については、『[Securing Users and Processes in Oracle Solaris 11.3](#)』の第 1 章「About Using Rights to Control Users and Processes」を参照してください。

2. `/etc/ldmp2v.conf` ファイルを作成し、次のデフォルトのプロパティを構成します。

- VDS – 仮想ディスクサービスの名前。VDS="primary-vds0" など
- VSW – 仮想スイッチの名前。VSW="primary-vsw0" など
- VCC – 仮想コンソール端末集配信装置の名前。VCC="primary-vcc0" など
- BACKEND_TYPE – zvol、file、または disk のバックエンドの種類

- **BACKEND_SPARSE** – バックエンドデバイスを空白のボリュームまたはファイルとして作成するか (**BACKEND_SPARSE="yes"**)、空白でないボリュームまたはファイルとして作成するか (**BACKEND_SPARSE="no"**) を決定します

- **BACKEND_PREFIX** – 仮想ディスクのバックエンドデバイスを作成する位置

BACKEND_TYPE="zvol" の場合、**BACKEND_PREFIX** 値を ZFS データセット名として指定します。**BACKEND_TYPE="files"** の場合、**BACKEND_PREFIX** 値は、/ からの相対的なディレクトリパス名として解釈されます。

たとえば、**BACKEND_PREFIX="tank/ldoms"** の場合、ZVOL は **tank/ldoms/domain-name** データセット、ファイルは **/tank/ldoms/domain-name** サブディレクトリに作成されます。

BACKEND_PREFIX プロパティは、**disk** バックエンドには適用されません。

- **BOOT_TIMEOUT** – Oracle Solaris OS ブート時のタイムアウト (秒)

詳細は、ダウンロード可能なバンドルに含まれている **ldmp2v.conf.sample** 構成ファイルを参照してください。

ldmp2v コマンドの使用

このセクションでは、変換の 3 つのフェーズの例が含まれます。

例 19-1 収集フェーズの例

ldmp2v collect コマンドの使用方法の例を次に示します。

- **NFS マウントされたファイルシステムを共有する。**次の例に、ソースシステムとターゲットシステムで NFS マウントされたファイルシステムを共有している場合に **collect** 手順を実行するもっとも簡単な方法を示します。

スーパーユーザーで、必要なすべての UFS ファイルシステムがマウントされていることを確認してください。

```
volumia# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c1t1d0s0    16516485  463289 15888032     3%    /
/proc                  0         0       0      0%    /proc
fd                     0         0       0      0%    /dev/fd
mnttab                 0         0       0      0%    /etc/mnttab
```

```

/dev/dsk/c1t1d0s3      8258597      4304 8171708      1%      /var
swap                   4487448          16 4487432      1%      /var/run
swap                   4487448          16 4487432      1%      /tmp
/dev/dsk/c1t0d0s0      1016122          9 955146      1%      /u01
vandikhout:/u1/home/dana
                        6230996752 1051158977 5179837775      17%      /home/dana

```

次の例に、ソースシステムとターゲットシステムが NFS マウントされたファイルシステムを共有している場合に収集ツールを実行する方法を示します。

```

volumia# ldmp2v collect -d /home/dana/volumia
Collecting system configuration ...
Archiving file systems ...
Determining which filesystems will be included in the archive...
Creating the archive...
895080 blocks
Archive creation complete.

```

- **NFS マウントされたファイルシステムを共有しない。**ソースシステムとターゲットシステムで、NFS マウントされたファイルシステムを共有していない場合、ファイルシステムイメージをローカルストレージに書き込んで、あとで制御ドメインにコピーできます。フラッシュアーカイブユーティリティーは、作成するアーカイブを自動的に除外します。

```

volumia# ldmp2v collect -d /var/tmp/volumia
Collecting system configuration ...
Archiving file systems ...
Determining which filesystems will be included in the archive...
Creating the archive...
895080 blocks
Archive creation complete.

```

フラッシュアーカイブと manifest ファイルを、/var/tmp/volumia ディレクトリからターゲットシステムにコピーします。

ヒント - 場合によっては、ldmp2v で cpio コマンドエラーが発生します。通常は、これらのエラーによって、「File size of etc/mnttab has increased by 435」などのメッセージが生成されます。ログファイルまたはシステムの状態を反映するファイルに関するメッセージは、無視してかまいません。すべてのエラーメッセージをよく確認してください。

- **ファイルシステムのバックアップ手順をスキップする。**すでに NetBackup などのサードパーティー製バックアップツールを使用してシステムのバックアップを作成してある場合、none アーカイブ方法を使用すればファイルシステムのバックアップ手順をスキップできます。このオプションを使用する場合、システム構成マニフェストのみが作成されます。

```
volumia# ldmp2v collect -d home/dana/p2v/volumia -a none
Collecting system configuration ...
The following file system(s) must be archived manually: / /u01 /var
```

-d で指定されたディレクトリがソースとターゲットのシステムで共有されない場合は、このディレクトリの内容を制御ドメインにコピーする必要があります。ディレクトリの内容は、準備フェーズの前に制御ドメインにコピーする必要があります。

例 19-2 準備フェーズの例

ldmp2v prepare コマンドの使用方法の例を次に示します。

- 次の例は、物理システムの MAC アドレスを保持しながら、/etc/ldmp2v.conf に構成されているデフォルトを使用することで、volumia という論理ドメインを作成します。

```
# ldmp2v prepare -d /home/dana/p2v/volumia -o keep-mac volumia
Creating vdisks ...
Creating file systems ...
Populating file systems ...
Modifying guest domain OS image ...
Removing SVM configuration ...
Unmounting guest file systems ...
Creating domain volumia ...
Attaching vdisks to domain volumia ...
```

- 次のコマンドは、volumia 論理ドメインに関する情報を表示します。

```
# ldm list -l volumia
```


NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
volumia	inactive	-----		2	4G		

NETWORK							
NAME	SERVICE	DEVICE	MAC	MODE	PVID	VID	
vnet0	primary-vsw0		00:03:ba:1d:7a:5a		1		

DISK						
NAME	DEVICE	TOUT	MPGROUP	VOLUME	SERVER	
disk0				volumia-vol0@primary-vds0		
disk1				volumia-vol1@primary-vds0		

- 次の例は、**-c** オプションを使用して、ドメインとそのバックエンドデバイスを完全に削除する方法を示しています。

```
# ldmp2v prepare -C volumia
Cleaning up domain volumia ...
Removing vdisk disk0 ...
Removing vdisk disk1 ...
Removing domain volumia ...
Removing volume volumia-vol0@primary-vds0 ...
Removing ZFS volume tank/ldoms/volumia/disk0 ...
Removing volume volumia-vol1@primary-vds0 ...
Removing ZFS volume tank/ldoms/volumia/disk1 ...
```

- 次の例は、**-m** オプションを使用してマウントポイントとその新しいサイズを指定することで、P2V の実行中に 1 つ以上のファイルシステムのサイズを変更する方法を示しています。

```
# ldmp2v prepare -d /home/dana/p2v/volumia -m /:8g volumia
Resizing file systems ...
Creating vdisks ...
Creating file systems ...
Populating file systems ...
Modifying guest domain OS image ...
Removing SVM configuration ...
Modifying file systems on SVM devices ...
Unmounting guest file systems ...
Creating domain volumia ...
Attaching vdisks to domain volumia ...
```

例 19-3 変換フェーズの例

ldmp2v convert コマンドの使用方法的例を次に示します。

- **ネットワークインストールサーバーを使用する。**ldmp2v convert コマンドは、指定した仮想ネットワークインタフェースを使用して、ネットワーク上でドメインをブートします。インストールサーバーで setup_install_server および add_install_client スクリプトを実行する必要があります。

Oracle Solaris 10 システムでは、Oracle Solaris JumpStart 機能を使用して、完全に自動化された変換を実行できます。この機能では、JumpStart サーバー上のクライアントに対して適切な sysidcfg およびプロファイルファイルを作成および構成する必要があります。プロファイルには次の行を含めるようにしてください。

```
install_type      upgrade
root_device       c0d0s0
```

sysidcfg ファイルは、アップグレード処理にのみ使用されます。したがって、次のような構成で十分であるはずです。

```
name_service=NONE
root_password=uQkoXlMLCsZhI
system_locale=C
timeserver=localhost
timezone=Europe/Amsterdam
terminal=vt100
security_policy=NONE
nfs4_domain=dynamic
auto_reg=disable
network_interface=PRIMARY {netmask=255.255.255.192
                             default_route=none protocol_ipv6=no}
```

JumpStart の使用の詳細については、『[Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#)』を参照してください。

注記 - sysidcfg ファイルの例に含まれる auto_reg キーワードは、Oracle Solaris 10 9/10 リリースで導入されたものです。このキーワードは、Oracle Solaris 10 9/10 以降のリリースを実行する場合にのみ必要です。

```
# ldmp2v convert -j -n vnet0 -d /p2v/volumia volumia
LDom volumia started
Waiting for Solaris to come up ...
Using Custom JumpStart
Trying 0.0.0.0...
Connected to 0.
Escape character is '^J'.

Connecting to console "volumia" in group "volumia" ....
Press ~? for control options ..
SunOS Release 5.10 Version Generic_137137-09 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.
Configuring devices.
Using RPC Bootparams for network configuration information.
Attempting to configure interface vnet0...
Configured interface vnet0
Reading ZFS config: done.
Setting up Java. Please wait...
Serial console, reverting to text install
Beginning system identification...
Searching for configuration file(s)...
Using sysid configuration file
    129.159.206.54:/opt/SUNWjet/Clients/volumia/sysidcfg
Search complete.
Discovering additional network configuration...
Completing system identification...
Starting remote procedure call (RPC) services: done.
System identification complete.
Starting Solaris installation program...
Searching for JumpStart directory...
Using rules.ok from 129.159.206.54:/opt/SUNWjet.
Checking rules.ok file...
Using begin script: Clients/volumia/begin
Using profile: Clients/volumia/profile
Using finish script: Clients/volumia/finish
Executing JumpStart preinstall phase...
```

```
Executing begin script "Clients/volumia/begin"...
Begin script Clients/volumia/begin execution completed.
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.
WARNING: Backup media not specified.  A backup media (backup_media)
        keyword must be specified if an upgrade with disk space reallocation
        is required

Processing profile

Loading local environment and services

Generating upgrade actions
Checking file system space: 100% completed
Space check complete.

Building upgrade script

Preparing system for Solaris upgrade

Upgrading Solaris: 10% completed
[...]
```

- **ISO イメージを使用する。**ldmp2v convert コマンドは、Oracle Solaris DVD ISO イメージを論理ドメインに接続し、そこからブートします。アップグレードを行うには、sysid のすべての質問に回答し、「Upgrade」を選択します。



注意 - ゲストドメインを変換する前に、安全性チェックが行われます。このチェックにより、元のシステムの IP アドレスのいずれもアクティブになっていなくて、ネットワーク上でアクティブな IP アドレスが重複しないことが確実にになります。-x skip-ping-test オプションを使用すると、この安全性チェックを省略できます。このチェックを省略すると、変換プロセスが速くなります。このオプションは、元のホストがアクティブになっていない場合など、重複する IP アドレスが存在しないことが確実な場合にのみ使用します。

sysid の質問への回答は、アップグレードプロセスの間のみ使用されます。このデータは、ディスク上の既存の OS イメージには適用されません。変換を実行するためのもっとも速く簡単な方法は、「Non-networked (ネットワークなし)」を選択することです。指定する root パスワードは、ソースシステムの root パスワードと一致している必要はありません。システムの元の ID はアップグレード時にも保持され、アップグレード以後のリブート時に有効になります。アップグレードの実行に必要な時間は、元のシステムにインストールされている Oracle Solaris Cluster によって異なります。

```
# ldmp2v convert -i /tank/iso/s10s_u5.iso -d /home/dana/p2v/volumia volumia
```

```
Testing original system status ...
```

```
LDom volumia started
```

```
Waiting for Solaris to come up ...
```

```
Select 'Upgrade' (F2) when prompted for the installation type.
```

```
Disconnect from the console after the Upgrade has finished.
```

```
Trying 0.0.0.0...
```

```
Connected to 0.
```

```
Escape character is '^]'.
```

```
Connecting to console "volumia" in group "volumia" ....
```

```
Press ~? for control options ..
```

```
Configuring devices.
```

```
Using RPC Bootparams for network configuration information.
```

```
Attempting to configure interface vnet0...
```

```
Extracting windowing system. Please wait...
```

```
Beginning system identification...
```

```
Searching for configuration file(s)...
```

```
Search complete.  
Discovering additional network configuration...  
Configured interface vnet0  
Setting up Java. Please wait...
```

Select a Language

- 0. English
- 1. French
- 2. German
- 3. Italian
- 4. Japanese
- 5. Korean
- 6. Simplified Chinese
- 7. Spanish
- 8. Swedish
- 9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help:

[...]

- Solaris Interactive Installation -----

This system is upgradable, so there are two ways to install the Solaris software.

The Upgrade option updates the Solaris software to the new release, saving as many modifications to the previous version of Solaris software as possible. Back up the system before using the Upgrade option.

The Initial option overwrites the system disks with the new version of Solaris software. This option allows you to preserve any existing file systems. Back up any modifications made to the previous version of Solaris software before starting the Initial option.

After you select an option and complete the tasks that follow, a summary of your actions will be displayed.

F2_Upgrade F3_Go Back F4_Initial F5_Exit F6_Help

Oracle VM Server for SPARC 物理から仮想への変換ツールの既知の問題

ldmp2v convert コマンド: ブート中の VxVM 警告メッセージ

Oracle Solaris 10 OS 上での Veritas Volume Manager (VxVM) 5.x の実行は、Oracle VM Server for SPARC P2V ツールでサポートされる (テストされた) 唯一のバージョンです。Solaris 8 および Solaris 9 オペレーティングシステム上で実行されている (3.x や 4.x などの) 古いバージョンの VxVM も動作する可能性があります。この場合は、ldmp2v convert コマンドを実行したあとの最初のブートで、VxVM ドライバからの警告メッセージが表示されることがあります。このメッセージは無視してかまいません。ゲストドメインがブートしたあと、古い VRTS* パッケージを削除できます。

```
Boot device: disk0:a File and args:
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright 1983-2009 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: normaal
Configuring devices.
/kernel/drv/sparcv9/vxdmp: undefined symbol 'romp'
WARNING: mod_load: cannot load module 'vxdmp'
WARNING: vxdmp: unable to resolve dependency, module 'misc/ted' not found
/kernel/drv/sparcv9/vxdmp: undefined symbol 'romp'
WARNING: mod_load: cannot load module 'vxdmp'
WARNING: vxdmp: unable to resolve dependency, module 'misc/ted' not found
/kernel/drv/sparcv9/vxio: undefined symbol 'romp'
WARNING: mod_load: cannot load module 'vxio'
WARNING: vxio: unable to resolve dependency, module 'drv/vxdmp' not found
WARNING: vxspec : CANNOT INITIALIZE vxio DRIVER
WARNING: VxVM vxspec V-5-0-0 vxspec: vxio not loaded. Aborting vxspec load
WARNING: vxspec : CANNOT INITIALIZE vxio DRIVER
WARNING: VxVM vxspec V-5-0-0 vxspec: vxio not loaded. Aborting vxspec load
WARNING: vxspec : CANNOT INITIALIZE vxio DRIVER
WARNING: VxVM vxspec V-5-0-0 vxspec: vxio not loaded. Aborting vxspec load
WARNING: vxspec : CANNOT INITIALIZE vxio DRIVER
WARNING: VxVM vxspec V-5-0-0 vxspec: vxio not loaded. Aborting vxspec load
WARNING: vxspec : CANNOT INITIALIZE vxio DRIVER
```

```
WARNING: VxVM vxspec V-5-0-0 vxspec: vxio not loaded. Aborting vxspec load
WARNING: vxspec : CANNOT INITIALIZE vxio DRIVER
NOTICE: VxVM not started
```

ldmp2v prepare -R の使用時にアップグレードオプションが提供されない

ルート (/) ファイルシステムを保持するスライスのパーティションタグが root に設定されていない場合、Oracle Solaris インストーラはアップグレードオプションを提供しません。この状況は、ゲストのブートディスクにラベル付けを行う際に、タグが明示的に設定されていない場合に発生します。format コマンドを使用して、次のようにパーティションタグを設定できます。

```
AVAILABLE DISK SELECTIONS:
0. c0d0 <SUN-DiskImage-10GB cyl 282 alt 2 hd 96 sec 768>
   /virtual-devices@100/channel-devices@200/disk@0
1. c4t2d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
   /pci@400/pci@0/pci@1/scsi@0/sd@2,0
2. c4t3d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
   /pci@400/pci@0/pci@1/scsi@0/sd@3,0
Specify disk (enter its number)[0]: 0
selecting c0d0
[disk formatted, no defect list found]
format> p

PARTITION MENU:
0      - change `0' partition
1      - change `1' partition
2      - change `2' partition
3      - change `3' partition
4      - change `4' partition
5      - change `5' partition
6      - change `6' partition
7      - change `7' partition
select - select a predefined table
modify - modify a predefined partition table
name   - name the current table
print  - display the current table
label  - write partition map and label to the disk
!<cmd> - execute <cmd>, then return
quit

partition> 0
Part      Tag      Flag      Cylinders      Size      Blocks
0 unassigned  wm        0              0          (0/0/0)      0

Enter partition id tag[unassigned]: root
Enter partition permission flags[wm]:
Enter new starting cyl[0]: 0
```



```
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 8g
partition> label
Ready to label disk, continue? y

partition>
```

ldmp2v コマンド: ufsdump のアーカイブ方法が使用されなくなった

ufsdump アーカイブを UFS ファイルシステム上のファイルによってバックアップされた仮想ディスク上に復元すると、システムがハングアップすることがあります。このような場合、ldmp2v prepare コマンドは終了します。この問題は、仮想ディスクが UFS ファイルシステム上のファイルであるときに、ldmp2v prepare -R /altroot コマンドの準備として ufsdump アーカイブを手動で復元した場合に発生することがあります。以前に作成された ufsdump アーカイブとの互換性のために、ldmp2v prepare コマンドを使用して、UFS ファイルシステム上のファイルによってバックアップされていない仮想ディスク上に ufsdump アーカイブを引き続き復元できます。ただし、ufsdump アーカイブの使用はお勧めできません。

◆◆◆ 第 20 章

電源管理の使用

この章では、Oracle VM Server for SPARC システムで電源管理を使用するための情報を提供します。

電源管理の使用

電源管理 (PM) を有効にするには、まず Oracle Integrated Lights Out Manager (ILOM) 3.0 ファームウェアで PM ポリシーを設定する必要があります。このセクションでは、Oracle VM Server for SPARC ソフトウェアで PM を使用するために必要な情報をまとめます。

ILOM の詳細については、次のドキュメントを参照してください。

- 『Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI 手順ガイド』の「消費電力のモニタリング」
- 『Oracle Integrated Lights Out Manager (ILOM) 3.0 機能更新およびリリースノート』

電源ポリシーは任意の時点でのシステムの電力使用量を管理します。ベースとなるプラットフォームに PM 機能が実装されていれば、次の電源ポリシーがサポートされます。

- **無効。**システムは、使用可能なすべての電力を使用できます。
- **パフォーマンス。**パフォーマンスにわずかな影響しか与えない次の PM 機能を 1 つ以上有効にします。
 - CPU コアの自動無効化
 - CPU クロックサイクルのスキップ
 - CPU の動的な電圧および周波数スケーリング (DVFS)
 - コヒーレンシリンクのスケーリング

- Oracle Solaris Power Aware Dispatcher (PAD)
- エラスティック。パフォーマンスのセクションで説明した PM 機能を使用して、現在の使用率レベルに合うようにシステムの電力使用量を調整します。たとえば、リソースの使用率が低いと、電力状態も引き下げられます。

電源管理システムの機能

PM の機能は次のとおりです。

- **CPU コアの自動的な無効化。**エラスティックまたはパフォーマンスのポリシーが有効になっている場合、CPU コアのすべてのハードウェアスレッド (ストランド) がドメインにバインドされていないときに、Logical Domains Manager によってそのコアが自動的に無効化されます。この機能は、UltraSPARC T2、UltraSPARC T2 Plus、SPARC T3、および SPARC T4 プラットフォームでのみ使用できます。
- **CPU クロックサイクルのスキップ。**エラスティックポリシーが有効な場合、Logical Domains Manager は、ドメインにバインドされている次の CPU リソースで命令を実行するクロックサイクル数を自動的に調整します。
 - プロセッサ (Oracle Solaris 10 または Oracle Solaris 11 OS を実行する SPARC T3 または SPARC T4)
 - コア (Oracle Solaris 10 OS を実行するドメイン上の SPARC M5 のみ)
 - コアのペア (Oracle Solaris 10 OS を実行するドメイン上の SPARC T5 または SPARC M6 のみ)
 - SPARC キャッシュクラスタ (SCC) (Oracle Solaris 10 OS を実行するドメイン上の SPARC T7 シリーズサーバーおよび SPARC M7 サーバーのみ)

プロセッサ、コア、コアのペア、または SCC にバインドされたストランドがない場合、Logical Domains Manager はサイクルスキッピングも適用します。

- **CPU の動的な電圧および周波数スケーリング (DVFS)。**エラスティックポリシーが有効な場合、Logical Domains Manager は、Oracle Solaris 10 OS を実行しているドメインにバインドされているプロセッサまたは SCC のクロック周波数を自動的に調整します。Logical Domains Manager は、バインドされたストランドがない SPARC T5、SPARC M5、および SPARC M6 プロセッサのクロック周波数も低下します。SPARC T7 シリーズサーバーでは、クロック周波数は SCC で削減されます。この機能は、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、および SPARC M7 シリーズサーバーでのみ使用できます。

- **コヒーレンシリンクのスケーリング**。エラスティックポリシーが有効な場合、Logical Domains Manager は、使用中のコヒーレンシリンク数を自動的に調整するようにハイパーバイザに指示します。この機能は、SPARC T5-2 システムでのみ使用できます。
- **電力の制限**。SPARC T3 サーバー、SPARC T4 サーバー、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、および SPARC M7 シリーズサーバーでは、システムの消費電力を制限するための**電力の制限**を設定できます。消費電力が電力の制限を超えている場合、PM は電力を削減する手法を使用します。ILOM サービスプロセッサ (SP) を使用して、電力の制限を設定できます。

次のドキュメントを参照してください。

- 『Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI 手順ガイド』
- 『Oracle Integrated Lights Out Manager (ILOM) 3.0 機能更新およびリリースノート』

ILOM インタフェースを使用して、電力の制限、猶予期間、および制限を超えた場合の動作を設定できます。電力の制限を超えた期間が猶予期間よりも長くなった場合、制限を超えた場合の動作が実行されます。

現在の消費電力が電力の制限を超えている場合、CPU の電力状態の引き下げが試みられます。消費電力が電力の制限以下に低下すると、それらのリソースの電力状態を引き上げることが許可されます。システムでエラスティックポリシーが有効にされている場合、リソースの電力状態の引き上げは使用レベルによって引き起こされます。

- **Solaris Power Aware Dispatcher (PAD)**。Oracle Solaris 11.1 OS を実行しているゲストドメインは、SPARC T5 サーバー、SPARC T7 シリーズサーバー、SPARC M5 サーバー、SPARC M6 サーバー、および SPARC M7 シリーズサーバー上で Power Aware Dispatcher (PAD) を使用して、アイドル状態のリソースや使用率が低いリソースの消費電力を最小限に抑えます。PAD は、Logical Domains Manager の代わりに、CPU または SCC クロックサイクルのスキップレベルおよび DVFS レベルを調整します。

ILOM 3.0 ファームウェアの CLI を使用して電源ポリシーを構成する手順については、『Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI 手順ガイド』の「消費電力のモニタリング」を参照してください。

消費電力データの表示

電源管理 (PM) 可観測性モジュールおよび `ldmpower` コマンドを使用すると、ドメインの CPU スレッドの消費電力データを表示できます。

`ldmd/pm_observability_enabled` サービス管理機能 (SMF) プロパティが `true` に設定されているため、PM 可観測性モジュールはデフォルトで有効にされています。[ldmd\(1M\)](#) マニュアルページを参照してください。

`ldmpower` コマンドには、消費電力レポートデータをカスタマイズできる次のオプションとオペランドがあります。

```
ldmpower [-ehprstvx | -o hours | -m minutes] | -c resource [-l domain-name[,domain-name[,...]]]
          [interval [count]]
```

オプションについては、[ldmpower\(1M\)](#) マニュアルページを参照してください。

このコマンドを非特権ユーザーとして実行するには、LDoms Power Mgmt Observability 権利プロファイルが割り当てられている必要があります。すでに LDoms Management または LDoms Review 権利プロファイルが割り当てられている場合、`ldmpower` コマンドを実行するためのアクセス権が自動的に付与されます。

Oracle VM Server for SPARC で権利がどのように使われているかについては、[29 ページの「Logical Domains Manager プロファイルの内容」](#)を参照してください。

これらの権利プロファイルは、ユーザー、またはその後ユーザーに割り当てられる役割に、直接割り当てることができます。これらのプロファイルのいずれかがユーザーに直接割り当てられている場合、CPU スレッドの消費電力データを表示するために `ldmpower` コマンドを正常に使用するには、`pfexec` コマンド、または `pfbash` または `pfksh` などのプロファイルシェルを使用する必要があります。[25 ページの「権利の使用による論理ドメインの管理の委任」](#)を参照してください。

次の例では、PM 可観測性モジュールを有効にする方法と、ドメインに割り当てられている CPU の消費電力データを収集する方法を示します。

例 20-1 電源管理可観測性モジュールの有効化

次のコマンドでは、`ldmd/pm_observability_enabled` プロパティが現在 `false` に設定されている場合に `true` に設定することによって、PM 可観測性モジュールを有効にします。

```
# svccfg -s ldmd setprop ldmd/pm_observability_enabled=true
# svcadm refresh ldmd
# svcadm restart ldmd
```

例 20-2 役割および権利プロファイルを使用したプロファイルシェルによる CPU スレッドの消費電力データの取得

- 次の例では、LDoms Power Mgmt Observability 権利プロファイルによって、ldmpower 役割を作成する方法を示します。この役割により、ldmpower コマンドの実行が許可されます。

```
primary# roleadd -P "LDoms Power Mgmt Observability" ldmpower
primary# passwd ldmpower
New Password:
Re-enter new Password:
passwd: password successfully changed for ldmpower
```

このコマンドは、sam ユーザーに ldmpower 役割を割り当てます。

```
primary# usermod -R ldmpower sam
```

ユーザー sam は ldmpower 役割を前提とし、ldmpower コマンドを使用できます。たとえば、次のように表示されます。

```
$ id
uid=700299(sam) gid=1(other)

$ su ldmpower
Password:

$ pfexec ldmpower
Processor Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 75         84         86
gdom1   47         24         19
gdom2   10         24         26
```

- 次の例は、権利プロファイルを使用して ldmpower コマンドを実行する方法を示しています。権利プロファイルをユーザーに割り当てます。

```
primary# usermod -P +"LDoms Power Mgmt Observability" sam
```

次のコマンドは、ユーザーが sam であり、All、Basic Solaris User、および LDoms Power Mgmt Observability 権利プロファイルが有効であることを確認する方法を示しています。

```
$ id
uid=702048(sam) gid=1(other)

$ profiles
```

```

All
Basic Solaris User
LDoms Power Mgmt Observability
$ pfexec ldmpower
Processor Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 75        84        86
gdom1   47        24        19
gdom2   10        24        26

```

例 20-3 プロセッサの消費電力データの表示

次の例に、ldmpower を使用して、ドメインのプロセッサ消費電力データを報告する方法を示します。

- 次のコマンドは、すべてのドメインについて、15 秒、30 秒、60 秒のプロセッサの移動平均消費電力データを表示します。

```

primary# ldmpower
Processor Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 75        84        86
gdom1   47        24        19
gdom2   10        24        26

```

- 次のコマンドは、すべてのドメイン (primary、gdom1、および gdom2) について、外挿消費電力データを表示します。

```

primary# ldmpower -x
System Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 585/57.47% 701/68.96% 712/70.22%
gdom1   132/12.97% 94/9.31%  94/9.30%
gdom2   298/29.27% 218/21.47% 205/20.22%

```

- 次のコマンドは、gdom2 および gdom5 ドメインについて、プロセッサの瞬間消費電力データを表示します。10 秒ごとに 5 回データを報告します。

```

primary# ldmpower -itl gdom2,gdom5 10 5
Processor Power Consumption in Watts
DOMAIN          TIMESTAMP          INSTANT

```


gdom2	2013.05.17 11:14:45	13
gdom5	2013.05.17 11:14:45	24
gdom2	2013.05.17 11:14:55	18
gdom5	2013.05.17 11:14:55	26
gdom2	2013.05.17 11:15:05	9
gdom5	2013.05.17 11:15:05	16
gdom2	2013.05.17 11:15:15	15
gdom5	2013.05.17 11:15:15	19
gdom2	2013.05.17 11:15:25	12
gdom5	2013.05.17 11:15:25	18

- 次のコマンドは、すべてのドメインについて、最近 12 時間の平均消費電力データを表示します。データは最後にリクエストされた 1 時間ごとの計算から 1 時間の間隔で表示されます。

```
primary# ldmpower -eto 12
```

Per domain MINIMUM and MAXIMUM power consumption ever recorded:

primary	2013.05.17 08:53:06	3	Min Processors
primary	2013.05.17 08:40:44	273	Max Processors
gdom1	2013.05.17 09:56:35	2	Min Processors
gdom1	2013.05.17 08:53:06	134	Max Processors
gdom2	2013.05.17 10:31:55	2	Min Processors
gdom2	2013.05.17 08:56:35	139	Max Processors

primary	2013.05.17 08:53:06	99	Min Memory
primary	2013.05.17 08:40:44	182	Max Memory
gdom1	2013.05.17 09:56:35	13	Min Memory
gdom1	2013.05.17 08:53:06	20	Max Memory
gdom2	2013.05.17 10:31:55	65	Min Memory
gdom2	2013.05.17 08:56:35	66	Max Memory

Processor Power Consumption in Watts

12 hour's worth of data starting from 2013.05.16 23:17:02

DOMAIN	TIMESTAMP	1 HOUR AVG

primary	2013.05.17 09:37:35	112
gdom1	2013.05.17 09:37:35	15
gdom2	2013.05.17 09:37:35	26
primary	2013.05.17 10:37:35	96
gdom1	2013.05.17 10:37:35	12
gdom2	2013.05.17 10:37:35	21
primary	2013.05.17 11:37:35	85
gdom1	2013.05.17 11:37:35	11
gdom2	2013.05.17 11:37:35	23
...		

◆◆◆ 第 21 章 21

Oracle VM Server for SPARC 管理情報ベース (MIB) ソフトウェアの使用

Oracle VM Server for SPARC 管理情報ベース (Management Information Base、MIB) を使用すると、サードパーティーのシステム管理アプリケーションによるドメインのリモートモニタリングと、簡易ネットワーク管理プロトコル (SNMP) による論理ドメイン (ドメイン) の起動および停止が可能になります。

Oracle VM Server for SPARC MIB ソフトウェアのインスタンスが 1 つだけ制御ドメインで実行できます。制御ドメインは、少なくとも Solaris 10 11/06 OS と Oracle VM Server for SPARC 2.2 ソフトウェアを実行するようにしてください。

この章では、次の項目について説明します。

- [475 ページの「Oracle VM Server for SPARC 管理情報ベースの概要」](#)
- [480 ページの「Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成」](#)
- [483 ページの「セキュリティの管理」](#)
- [484 ページの「ドメインのモニタリング」](#)
- [508 ページの「SNMP トラップの使用」](#)
- [515 ページの「ドメインの起動と停止」](#)

Oracle VM Server for SPARC 管理情報ベースの概要

このセクションでは、次のトピックについて説明します。

- [476 ページの「関連の製品および機能」](#)
- [476 ページの「ソフトウェアコンポーネント」](#)
- [477 ページの「システム管理エージェント」](#)

- [478 ページの「Logical Domains Manager と Oracle VM Server for SPARC MIB」](#)
- [479 ページの「Oracle VM Server for SPARC MIB オブジェクトツリー」](#)

関連の製品および機能

Oracle VM Server for SPARC MIB を使用するには、次のソフトウェア製品と機能の使用方法を理解する必要があります。

- Oracle Solaris OS
- Oracle VM Server for SPARC ソフトウェア
- SNMP
- SNMP 管理情報ベース (MIB)
- Oracle Solaris SNMP エージェント
- SNMP バージョン 1 (SNMPv1)、SNMP バージョン 2 (SNMPv2c)、および SNMP バージョン 3 (SNMPv3) のプロトコル
- 管理情報構造 (SMI) バージョン 1 およびバージョン 2
- 管理情報ベース (MIB) の構造
- 抽象構文記法 (ASN.1)

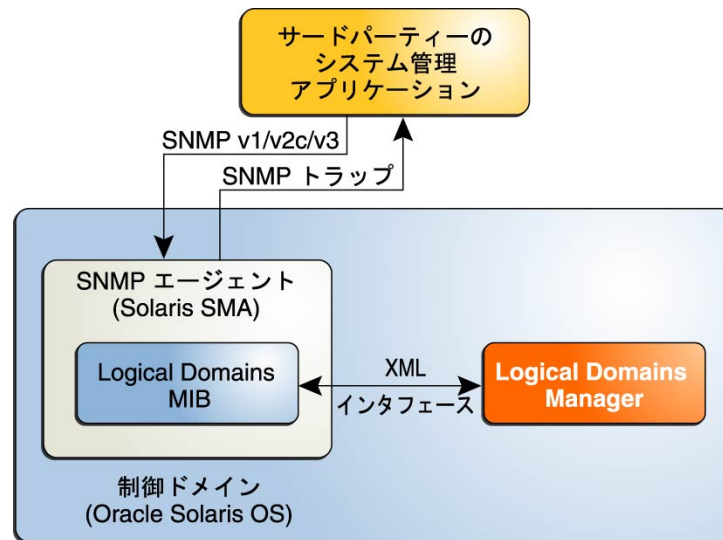
ソフトウェアコンポーネント

Oracle VM Server for SPARC MIB パッケージ `SUNWldmib.v` には、次のソフトウェアコンポーネントが含まれています。

- `SUN-LDOM-MIB.mib` はテキストファイル書式の SNMP 管理情報ベースです。このファイルは Oracle VM Server for SPARC MIB のオブジェクトを定義します。
- `ldomMIB.so` は共用ライブラリ形式のシステム管理エージェント拡張モジュールです。このモジュールは、Oracle Solaris SNMP エージェントが Oracle VM Server for SPARC MIB で指定された情報リクエストに応答し、トラップを生成できるようにします。

次の図は、Oracle VM Server for SPARC MIB、Oracle Solaris SNMP エージェント、Logical Domains Manager、およびサードパーティーのシステム管理アプリケーションの間のやり取りを示しています。この図に示されているやり取りは、[477 ページの「システム管理エージェント」](#)と[478 ページの「Logical Domains Manager と Oracle VM Server for SPARC MIB」](#)で説明しています。

図 21-1 Oracle Solaris SNMP エージェント、Logical Domains Manager およびサードパーティーのシステム管理アプリケーションとの Oracle VM Server for SPARC MIB のやり取り



システム管理エージェント

Oracle Solaris SNMP エージェントは次の機能を実行します。

- サードパーティーのシステム管理アプリケーションからの要求を待機し、Oracle VM Server for SPARC MIB が提供するデータを取得または設定します。エージェントは標準の SNMP ポート 161 で待機します。
- SNMP 通知の標準ポート 162 を使用して、構成済みのシステム管理アプリケーションにトラップを発行します。

Oracle VM Server for SPARC MIB は、制御ドメイン上の Oracle Solaris OS のデフォルト Oracle Solaris SNMP エージェントによってエクスポートされます。

Oracle Solaris SNMP エージェントは、SNMP バージョン v1、v2c、および v3 の取得、設定、およびトラップ機能をサポートします。ほとんどの Oracle VM Server for SPARC MIB オブジェクトは、モニタリングを目的とする読み取り専用です。ただし、ドメインを起動または停

止するには、ldomTable テーブルの ldomAdminState プロパティに値を書き込む必要があります。[表21-1「ドメインテーブル \(ldomTable\)」](#)を参照してください。

Logical Domains Manager と Oracle VM Server for SPARC MIB

ドメインは、ゲストオペレーティングシステムの一連の仮想リソースで構成されるコンテナです。Logical Domains Manager は、ドメインを作成、構成、および管理するためのコマンド行インタフェース (CLI) を提供します。Logical Domains Manager と Oracle VM Server for SPARC MIB は次の仮想リソースをサポートします。

- CPU
- メモリー
- ディスク、ネットワーク、およびコンソール I/O
- 暗号化装置

XML ベースの制御インタフェースの解析

Logical Domains Manager は XML ベースの制御インタフェースを Oracle VM Server for SPARC MIB にエクスポートします。Oracle VM Server for SPARC MIB は XML インタフェースを解析し、MIB にデータを取り込みます。Oracle VM Server for SPARC MIB は制御ドメインにサポートを提供するだけです。

SNMP トラップの提供

Oracle VM Server for SPARC MIB は、更新とステータス変更のために Logical Domains Manager を定期的にポーリングし、SNMP トラップをシステム管理アプリケーションに発行します。

障害と復旧の情報の提供

Oracle VM Server for SPARC MIB が必要なリソースを割り当てられなくなった場合、管理情報ベースは、SNMP エージェントを介して、システム管理アプリケーションに一般的なエ

ラーを返します。SNMPトラップ配信メカニズムはエラーを確認しません。Oracle VM Server for SPARC MIB には、特定の状態やチェックポイントは実装されていません。Oracle VM Server for SPARC MIB のある Oracle Solaris SNMP エージェントは、init プロセスとサービス管理機能 (SMF) によって起動およびモニターされます。Oracle Solaris SNMP エージェントが失敗して終了した場合、SMF が自動的にプロセスを再起動し、新しいプロセスが Oracle VM Server for SPARC MIB モジュールを動的に再起動します。

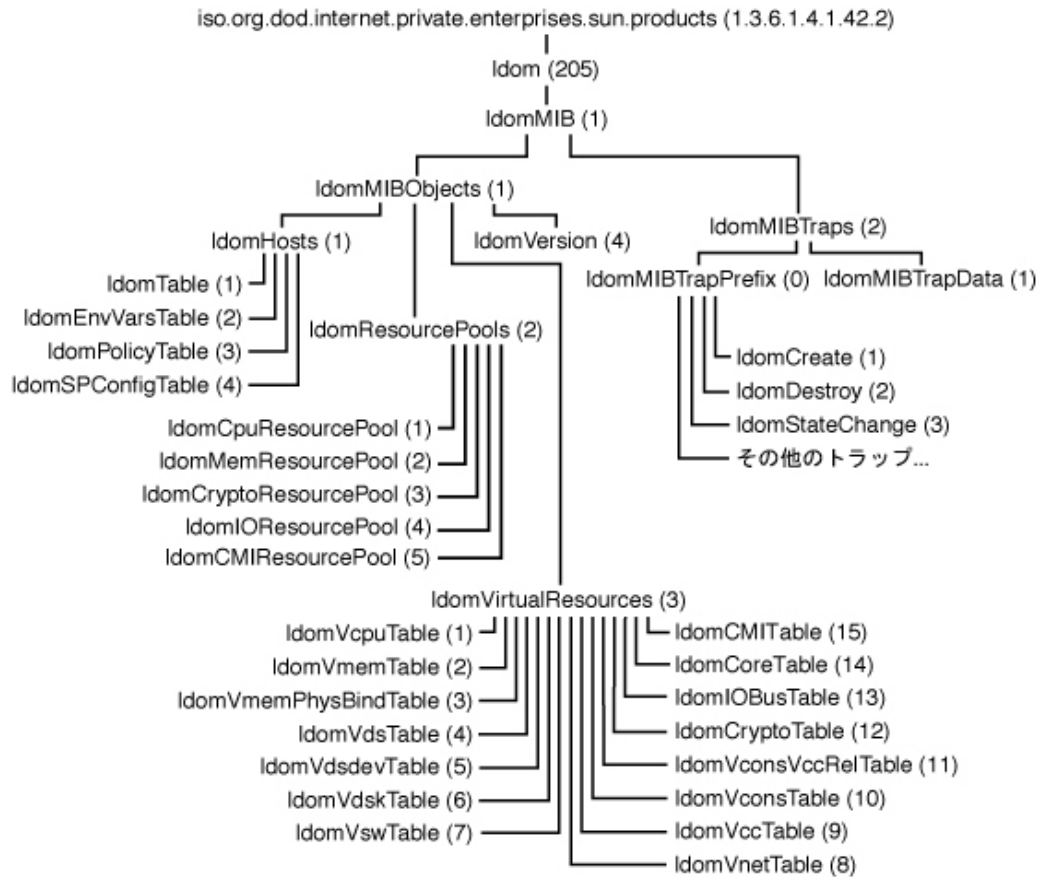
Oracle VM Server for SPARC MIB オブジェクトツリー

SNMP 管理対象オブジェクトは、ツリー型の階層に編成されます。オブジェクト識別子 (OID) は、ツリーのノードに基づいた一連の整数で構成され、ドットで区切られます。それぞれの管理対象オブジェクトは、数字で表わされた OID のほか、関連付けられたテキスト形式の名前を持っています。Oracle VM Server for SPARC MIB は、オブジェクトツリーの次の部分で `ldom (205)` ブランチとして登録されます。

```
iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).sun(42).products(2)
```

次の図は、Oracle VM Server for SPARC MIB 配下の主要なサブツリーを示しています。

図 21-2 Oracle VM Server for SPARC MIB ツリー



Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成

このセクションでは、Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成について説明します。SNMP の管理については、`snmpd.conf(4)` または `snmpd.conf(5)` のマニュアルページを参照してください。

Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成

次の表は、Oracle VM Server for SPARC MIB ソフトウェアのインストールおよび構成に使用できるタスクを示しています。

タスク	説明	手順の参照先
Oracle VM Server for SPARC MIB ソフトウェアパッケージをprimaryドメインにインストールします。	pkg install コマンドを使用して、system/ldoms/mib パッケージをインストールします。	481 ページの「Oracle VM Server for SPARC MIB ソフトウェアパッケージのインストール方法」
Oracle VM Server for SPARC MIB に問い合わせるために、Oracle VM Server for SPARC MIB モジュールを Oracle Solaris SNMP エージェントにロードします。	ldomMIB.so モジュールをロードするように、SNMP 構成ファイルを変更します。	482 ページの「Oracle VM Server for SPARC MIB モジュールを Oracle Solaris SNMP エージェントにロードする方法」
primaryドメインから Oracle VM Server for SPARC MIB ソフトウェアパッケージを削除します。	pkg remove コマンドを使用して、system/ldoms/mib パッケージを削除します。	482 ページの「Oracle VM Server for SPARC MIB ソフトウェアパッケージを削除する方法」

▼ Oracle VM Server for SPARC MIB ソフトウェアパッケージのインストール方法

この手順では、Oracle VM Server for SPARC MIB ソフトウェアパッケージをインストールする方法について説明します。Oracle VM Server for SPARC MIB ソフトウェアパッケージは、Oracle VM Server for SPARC 3.3 ソフトウェアの一部として提供されています。

Oracle VM Server for SPARC MIB パッケージには次のファイルが含まれています。

- /opt/SUNWldmib/lib/mibs/SUN-LDOM-MIB.mib
- /opt/SUNWldmib/lib/ldomMIB.so

始める前に Oracle VM Server for SPARC 3.3 ソフトウェアをダウンロードしてインストールします。『[Oracle VM Server for SPARC 3.3 インストールガイド](#)』の第 2 章「ソフトウェアのインストール」を参照してください。

- Oracle VM Server for SPARC MIB ソフトウェアパッケージ system/ldoms/mib をインストールします。

```
primary# pkg install -v -g IPS-package-directory/ldoms.repo mib
```

次の手順 このパッケージのインストール後、システムを構成して、Oracle VM Server for SPARC MIB モジュールを動的にロードできます。[482 ページの「Oracle VM Server for SPARC MIB モジュールを Oracle Solaris SNMP エージェント にロードする方法」](#)を参照してください。

▼ Oracle VM Server for SPARC MIB モジュールを Oracle Solaris SNMP エージェント にロードする方法

Oracle VM Server for SPARC MIB に問い合わせるには、Oracle VM Server for SPARC MIB モジュール `ldomMIB.so` を Oracle Solaris SNMP エージェント にロードする必要があります。Oracle VM Server for SPARC MIB モジュールは動的にロードされるので、エージェントバイナリの再コンパイルおよび再リンクの必要なしに、モジュールが SNMP エージェントに含まれます。

この手順では、Oracle VM Server for SPARC MIB モジュールを動的にロードするようにシステムを構成する方法を説明します。Oracle Solaris SNMP エージェント を再起動せずにモジュールを動的にロードする手順は、『*Solaris システム管理エージェント開発者ガイド*』で説明しています。Oracle Solaris SNMP エージェント の詳細は、『*Solaris システム管理ガイド*』を参照してください。

1. SNMP 構成ファイルを更新します。

次の行を `/etc/net-snmp/snmp/snmpd.conf` 構成ファイルに追加します。

```
dload ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

2. SMF サービスを再起動します。

```
primary# svcadm restart svc:/application/management/net-snmp:default
```

▼ Oracle VM Server for SPARC MIB ソフトウェアパッケージを削除する方法

この手順では、Oracle VM Server for SPARC MIB ソフトウェアパッケージを削除し、Oracle VM Server for SPARC MIB モジュールをアンロードする方法について説明します。

1. SMF サービスを停止します。

```
primary# svcadm disable svc:/application/management/net-snmp:default
```

2. **primary**ドメインから Oracle VM Server for SPARC MIB ソフトウェアパッケージを削除します。

```
primary# pkg uninstall system/ldoms/mib
```

3. **SNMP 構成ファイルを更新します。**

インストール中に /etc/net-snmp/snmp/snmpd.conf ファイルに追加した行を削除します。

```
dlmod ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

4. **SMF サービスを再起動します。**

```
primary# svcadm restart svc:/application/management/net-snmp:default
```

セキュリティの管理

このセクションでは、新しい Simple Network Management Protocol (SNMP) バージョン 3 (v3) ユーザーを作成し、セキュアなアクセスを Oracle Solaris SNMP エージェントに提供する方法について説明します。SNMP バージョン 1 (v1) およびバージョン 2 (v2c) では、アクセス制御メカニズムはコミュニティ文字列で、SNMP サーバーとそのクライアントの間の関係を定義します。この文字列は、システムへのユーザーのアクセスを制御するパスワードと同じように、サーバーへのクライアントアクセスを制御します。『*Solaris システム管理エージェント管理ガイド*』を参照してください。

注記 - snmpv3 ユーザーを作成すると、Oracle VM Server for SPARC MIB において SNMP で Oracle Solaris SNMP エージェントを使用できるようになります。このタイプのユーザーは、Logical Domains Manager 向けの Oracle Solaris 権利機能を使用して構成された可能性のあるユーザーと通信したり競合したりしません。

▼ 最初の snmpv3 ユーザーを作成する方法

この手順では、最初の snmpv3 ユーザーを作成する方法を説明します。

この最初のユーザーを複製することにより、追加ユーザーを作成できます。複製すると、続いて作成されるユーザーが最初のユーザーの認証タイプとセキュリティタイプを継承できます。これらのタイプはあとで変更できます。

最初のユーザーを複製するときに、その新しいユーザーの秘密鍵データを設定します。最初のユーザーと続いて構成されるユーザーのパスワードを知っておく必要があります。最初のユーザーから一度に 1 人のユーザーだけを複製できます。Oracle Solaris OS のバージョンに対応した『Solaris システム管理エージェント管理ガイド』の「セキュリティー保護された追加 SNMPv3 ユーザーを作成するには」を参照してください。

1. Oracle Solaris SNMP エージェント を停止します。

```
primary# svcadm disable svc:/application/management/net-snmp:default
```

2. 最初のユーザーを作成します。

この段階によって、選択したパスワード *my-password* を持つユーザー *initial-user* が作成され、エントリが */etc/sma/snmp/snmpd.conf* ファイルに追加されます。このエントリによって、最初のユーザーにエージェントへの読み取りアクセスと書き込みアクセスが与えられます。

注記 - パスワードは、8 文字以上である必要があります。

```
primary# /usr/bin/net-snmp-config --create-snmpv3-user -a my-password initial-user
```

3. Oracle Solaris SNMP エージェント を起動します。

```
primary# svcadm enable svc:/application/management/net-snmp:default
```

4. 最初のユーザーが作成されていることを確認します。

```
primary# snmpget -v 3 -u initial-user -l authNoPriv -a MD5 \
-A my-password localhost sysUpTime.0
```

ドメインのモニタリング

このセクションでは、Oracle VM Server for SPARC MIB に問い合わせることによって論理ドメイン (ドメイン) をモニターする方法を説明します。このセクションではまた、さまざまなタイプの MIB について説明します。

このセクションでは、次のトピックについて説明します。

- [485 ページの「環境変数の設定」](#)
- [485 ページの「Oracle VM Server for SPARC MIB の問い合わせ」](#)

- [488 ページの「Oracle VM Server for SPARC MIB 情報の取得」](#)

環境変数の設定

Oracle VM Server for SPARC MIB に問い合わせる前に、PATH、MIBDIRS、および MIBS 環境変数を設定する必要があります。

```
$ PATH=/usr/bin:$PATH; export PATH
$ MIBDIRS=/opt/SUNWldmib/lib/mibs:/etc/net-snmp/snmp/mibs; export MIBDIRS
$ MIBS+=SUN-LDOM-MIB; export MIBS
```

Oracle VM Server for SPARC MIB の問い合わせ

システムに多数のドメインがある場合、SNMP 要求への応答が可能になる前に、SNMP エージェントがタイムアウトする可能性があります。タイムアウト値を増やすには、`-t` オプションを使用して、より長いタイムアウト値を指定します。たとえば、次の `snmpwalk` コマンドはタイムアウト値を 20 秒に設定します。

```
# snmpwalk -t 20 -v1 -c public localhost SUN-LDOM-MIB::ldomTable
```

また、`-t` オプションを使用して、`snmpget` コマンドおよび `snmptable` コマンドのタイムアウト値も指定できます。

- 単一の MIB オブジェクトを取得するには:

```
# snmpget -v version -c community-string host MIB-object
```

- 多数の MIB オブジェクトを取得するには:

`snmpwalk` コマンドまたは `snmptable` コマンドを使用します。

```
# snmpwalk -v version -c community-string host MIB-object
```

```
# snmptable -v version -c community-string host MIB-object
```

注記 - `snmptable` コマンドを `-v2c` または `-v3` オプションを付けて使用して Oracle VM Server for SPARC MIB 2.1 ソフトウェアをクエリーすると、空の SNMP テーブルが返されます。`snmptable` コマンドで `-v1` オプションを使用すると、期待どおりに動作します。

この問題を回避するには、`-CB` オプションを使用し、GETBULK リクエストではなく GETNEXT リクエストのみを使用してデータを取得します。[485 ページの「Oracle VM Server for SPARC MIB の問い合わせ」](#)を参照してください。

例 21-1 単一の Oracle VM Server for SPARC MIB オブジェクトの取得 (snmpget)

次の snmpget コマンドは ldomVersionMajor オブジェクトの値を問い合わせます。このコマンドは、ホスト localhost に snmpv1 (-v1) とコミュニティ文字列 (-c public) を指定します。

```
# snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMajor.0
SUN-LDOM-MIB::ldomVersionMajor.0 = INTEGER: 1
```

例 21-2 ldomTable からのオブジェクト値の取得 (snmpwalk)

次の例は、snmpwalk コマンドを使用して ldomTable からオブジェクト値を取得する方法を示しています。

- 次の snmpwalk -v1 コマンドは、ldomTable テーブルのすべてのオブジェクトの値を返します。

```
# snmpwalk -v1 -c public localhost SUN-LDOM-MIB::ldomTable
SUN-LDOM-MIB::ldomName.1 = STRING: primary
SUN-LDOM-MIB::ldomName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldomAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldomAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldomOperState.2 = INTEGER: bound(6)
SUN-LDOM-MIB::ldomNumVCpu.1 = INTEGER: 8
SUN-LDOM-MIB::ldomNumVCpu.2 = INTEGER: 4
SUN-LDOM-MIB::ldomMemSize.1 = INTEGER: 3360
SUN-LDOM-MIB::ldomMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldomMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomNumCrypto.1 = INTEGER: 1
SUN-LDOM-MIB::ldomNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldomNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldomNumIOBus.2 = INTEGER: 0
SUN-LDOM-MIB::ldomUUID.1 = STRING: 5f8817d4-5d2e-6f7d-c4af-91b5b34b5723
SUN-LDOM-MIB::ldomUUID.2 = STRING: 11284146-87ca-4877-8d80-cd0f60d5ec26
SUN-LDOM-MIB::ldomMacAddress.1 = STRING: 00:14:4f:46:47:d6
SUN-LDOM-MIB::ldomMacAddress.2 = STRING: 00:14:4f:f8:d5:6c
SUN-LDOM-MIB::ldomHostID.1 = STRING: 0x844647d6
SUN-LDOM-MIB::ldomHostID.2 = STRING: 0x84f8d56c
```

```

SUN-LDOM-MIB::ldomFailurePolicy.1 = STRING: ignore
SUN-LDOM-MIB::ldomFailurePolicy.2 = STRING: ignore
SUN-LDOM-MIB::ldomMaster.1 = STRING:
SUN-LDOM-MIB::ldomMaster.2 = STRING:
SUN-LDOM-MIB::ldomExtMapinSpace.1 = STRING: off
SUN-LDOM-MIB::ldomExtMapinSpace.2 = STRING: off
SUN-LDOM-MIB::ldomWholeCore.1 = INTEGER: 0
SUN-LDOM-MIB::ldomWholeCore.2 = INTEGER: 0
SUN-LDOM-MIB::ldomCpuArch.1 = STRING: native
SUN-LDOM-MIB::ldomCpuArch.2 = STRING: native
SUN-LDOM-MIB::ldomShutdownGroup.1 = INTEGER: 0
SUN-LDOM-MIB::ldomShutdownGroup.2 = INTEGER: 15
SUN-LDOM-MIB::ldomPerfCounters.1 = STRING: htstrand
SUN-LDOM-MIB::ldomPerfCounters.2 = STRING: global,htstrand
SUN-LDOM-MIB::ldomNumCMI.1 = INTEGER: 0
SUN-LDOM-MIB::ldomNumCMI.2 = INTEGER: 0

```

- 次の `snmpwalk` コマンドは、`snmpv2c` と `snmpv3` を使用して、`ldomTable` の内容を取得します。

```

# snmpwalk -v2c -c public localhost SUN-LDOM-MIB::ldomTable
# snmpwalk -v 3 -u test -l authNoPriv -a MD5 -A testpassword localhost \
SUN-LDOMMIB::ldomTable

```

例 21-3 `ldomTable` からのオブジェクト値の表形式での取得 (`snmptable`)

次の例は、`snmptable` コマンドを使用して `ldomTable` からオブジェクト値を表形式で取得する方法を示しています。

- 次の `snmptable -v1` コマンドは、`ldomTable` の内容を表形式で表示します。

```
# snmptable -v1 -c public localhost SUN-LDOM-MIB::ldomTable
```

- 次の `snmptable` コマンドは、`snmpv2c` を使用して、`ldomTable` の内容を表形式で表示します。

`v2c` または `v3 snmptable` コマンドの場合、`-CB` オプションを使用して、`GETBULK` ではなく `GETNEXT` 要求だけを指定し、データを取得することに注意してください。

```
# snmptable -v2c -CB -c public localhost SUN-LDOM-MIB::ldomTable
```

Oracle VM Server for SPARC MIB 情報の取得

このセクションでは、Oracle VM Server for SPARC MIB から表またはスカラーオブジェクトの形式で取得できる情報について説明します。

ドメインテーブル (ldomTable)

ldomTable は、システムの各ドメインを表すときに使用されます。情報には、仮想 CPU、メモリー、暗号化装置、および I/O バスのリソース制約が含まれます。このテーブルには、汎用一意識別子 (UUID)、MAC アドレス、ホスト ID、障害ポリシー、マスタードメインなどのドメイン情報が含まれます。

表 21-1 ドメインテーブル (ldomTable)

名前	データ型	アクセス	説明
ldomIndex	整数	アクセス不可	このテーブルのインデックスとして使用される整数
ldomName	表示文字列	読み取り専用	ドメインの名前
ldomAdminState	整数	読み取り/書き込み	アクティブな管理のために、ドメインの起動や停止を実行します。 ■ 値 1 はドメインを起動します ■ 値 2 はドメインを停止します
ldomOperState	整数	読み取り専用	ドメインの現在の状態。次の値のいずれか 1 つです。 ■ 1 はアクティブ状態 ■ 2 は停止状態 ■ 3 はアクティブでない状態 ■ 4 はバインド状態 ■ 5 はアンバインド状態 ■ 6 はバウンド状態 ■ 7 は起動状態
ldomNumVCPU	整数	読み取り専用	使用している仮想 CPU の数。ドメインがアクティブでない状態の場合、この値は要求された仮想 CPU の数になります。
ldomMemSize	整数	読み取り専用	使用している仮想メモリーの容量。ドメインがアクティブでない状態の場合、この値は要求された記憶域サイズになります。

名前	データ型	アクセス	説明
ldomMemUnit	整数	読み取り専用	次のメモリー単位のいずれか 1 つです。 ■ 1 は KB ■ 2 は MB ■ 3 は GB ■ 4 は バイト 指定されていない場合、単位の値はバイトです。
ldomNumCrypto	整数	読み取り専用	使用している暗号化装置の数。ドメインがアクティブでない状態の場合、この値は要求された暗号化装置の数になります。
ldomNumIOBus	整数	読み取り専用	使用している物理 I/O デバイスの数
ldomUUID	表示文字列	読み取り専用	ドメインの UUID
ldomMacAddress	表示文字列	読み取り専用	ドメインの MAC アドレス
ldomHostID	表示文字列	読み取り専用	ドメインのホスト ID
ldomFailurePolicy	表示文字列	読み取り専用	マスタートメインの障害ポリシー。ignore、panic、reset、stop のいずれかです。
ldomMaster	表示文字列	読み取り専用	1 つのスレーブドメインに対する、最大 4 つのマスタートメインの名前
ldomExtMapinSpace	表示文字列	読み取り専用	ドメインの拡張マップイン領域。拡張されたマップイン領域は、追加の LCD 共有メモリー空間を参照します。このメモリー空間は、直接マップされた共有メモリーを使用する多数の仮想 I/O デバイスをサポートするために必要となります。この領域は、パフォーマンスとスケーラビリティを向上させるために仮想ネットワークデバイスによっても使用されます。デフォルト値は off です。
ldomWholeCore	整数	読み取り専用	コア全体のみを使用するようにドメインを制約します。コア全体の制約が有効になっていない場合、値は 0 です。それ以外の場合、値は最大のコア数を示します。
ldomCpuArch	表示文字列	読み取り専用	ドメインの CPU アーキテクチャー。CPU アーキテクチャーは、ドメインを別の sun4v CPU アーキテクチャーに移行できるかどうかを指定します。有効な値は次のとおりです。 ■ native、同じ sun4v CPU アーキテクチャーのプラットフォームへのドメイン移行のみが

名前	データ型	アクセス	説明
			許可されていることを意味します (デフォルト値) generic、互換性のあるすべての sun4v CPU アーキテクチャーへのドメイン移行が許可されていることを意味します
ldomShutdownGroup	整数	読み取り専用	ゲストドメインの停止グループ番号。Fujitsu M10 サーバーでは、SP で開始された順序付けられた停止により、ドメインがその停止グループ番号の降順 (15 から 0 への順) で停止されます。デフォルト値は 15 です。
ldomPerfCounters	文字列	読み取り専用	ゲストドメインのパフォーマンスレジスタアクセス情報。値は global (一度に 1 つのドメインでのみ) およびオプションで htstrand または strand のいずれかを指定できます。デフォルト値は htstrand です。
ldomNumCMI	整数	読み取り専用	使用された CMI リソースの数。ドメインが非アクティブな状態にある場合、この値は CMI リソースの要求された数です。

環境変数テーブル (ldomEnvVarsTable)

ldomEnvVarsTable は、すべてのドメインが使用する OpenBoot PROM 環境変数を示します。

表 21-2 環境変数テーブル (ldomEnvVarsTable)

名前	データ型	アクセス	説明
ldomEnvVarsLdomIndex	整数	読み取り専用	OpenBoot PROM 環境変数を含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomEnvVarsIndex	整数	読み取り専用	このテーブルの OpenBoot PROM 環境変数にインデックスを作成するために使用される整数
ldomEnvVarsName	表示文字列	読み取り専用	OpenBoot PROM 変数の名前
ldomEnvVarsValue	表示文字列	読み取り専用	OpenBoot PROM 変数の値

ドメインポリシーテーブル (ldomPolicyTable)

ldomPolicyTable は、すべてのドメインに適用される動的リソース管理 (DRM) ポリシーを示します。

表 21-3 ドメインポリシーテーブル (ldomPolicyTable)

名前	データ型	アクセス	説明
ldomPolicyLdomIndex	整数	読み取り専用	DRM ポリシーを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomPolicyIndex	整数	アクセス不可	このテーブルの DRM ポリシーにインデックスを作成するために使用される整数
ldomPolicyName	表示文字列	読み取り専用	ポリシー名
ldomPolicyStatus	表示文字列	読み取り専用	ポリシーステータス
ldomPolicyPriority	整数	読み取り専用	ポリシーがオーバーラップするとき、選択する DRM ポリシーを決定するために使用する優先度
ldomPolicyVcpuMin	整数	読み取り専用	ドメインの仮想 CPU の最小数
ldomPolicyVcpuMax	整数	読み取り専用	ドメインの仮想 CPU の最大数。unlimited の値は最大整数値 2147483647 を使用します。
ldomPolicyUtilLower	整数	読み取り専用	ポリシー分析がトリガーされる使用率の下限レベル
ldomPolicyUtilUpper	整数	読み取り専用	ポリシー分析がトリガーされる使用率の上限レベル
ldomPolicyTodBegin	表示文字列	読み取り専用	hh:mm:[ss] の形式で表示される、ポリシーの実効開始時間
ldomPolicyTodEnd	表示文字列	読み取り専用	hh:mm:[ss] の形式で表示される、ポリシーの実効停止時間
ldomPolicySampleRate	整数	読み取り専用	リソースサイクル時間 (秒単位)
ldomPolicyElasticMargin	整数	読み取り専用	仮想 CPU の数を減らした場合の変動を回避するための、util-lower プロパティ (ldomPolicyUtilLower) と使用していない仮想 CPU 数の間のバッファ量
ldomPolicyAttack	整数	読み取り専用	いずれか 1 つのリソース制御サイクル中に追加するリソースの最大量。unlimited の値は最大整数値 2147483647 を使用します。
ldomPolicyDecay	整数	読み取り専用	いずれか 1 つのリソース制御サイクル中に削除するリソースの最大量

サービスプロセッサ構成テーブル (ldomSPConfigTable)

ldomSPConfigTable は、すべてのドメインのサービスプロセッサ (SP) 構成を示します。

表 21-4 サービスプロセッサ構成テーブル (ldomSPConfigTable)

名前	データ型	アクセス	説明
ldomSPConfigIndex	整数	アクセス不可	このテーブルの SP 構成にインデックスを作成するために使用される整数
ldomSPConfigName	表示文字列	読み取り専用	SP 構成名
ldomSPConfigStatus	表示文字列	読み取り専用	SP 構成ステータス

ドメインリソースプールとスカラー変数

次のリソースをドメインに割り当てられます。

- 仮想 CPU (vcpu)
- メモリー (mem)
- 暗号化装置 (mau)
- 仮想スイッチ (vsw)
- 仮想ネットワーク (vnet)
- 仮想ディスクサーバー (vds)
- 仮想ディスクサーバーデバイス (vdsdev)
- 仮想ディスク (vdisk)
- 仮想コンソール端末集配信装置 (vcc)
- 仮想コンソール (vcons)
- 物理 I/O デバイス (io)
- CMI リソース (cmi)

次のスカラー MIB 変数は、リソースプールとそれらのプロパティを表すために使用します。

表 21-5 CPU リソースプールのスカラー変数

名前	データ型	アクセス	説明
ldomCpuRpCapacity	整数	読み取り専用	ldomCpuRpCapacityUnit のリソースプールが許可する最大予約数

名前	データ型	アクセス	説明
ldomCpuRpReserved	整数	読み取り専用	リソースプールから現在予約されている CPU のプロセッサクロック速度 (MHz) の累積
ldomCpuRpCapacityUnit と ldomCpuRpReservedUnit	整数	読み取り専用	次の CPU 割り当て単位のいずれか 1 つです。 ■ 1 は MHz ■ 2 は GHz デフォルト値は MHz です。

表 21-6 メモリリソースプールのスカラー変数

名前	データ型	アクセス	説明
ldomMemRpCapacity	整数	読み取り専用	MemRpCapacityUnit のリソースプールが許可する最大予約数
ldomMemRpReserved	整数	読み取り専用	リソースプールから現在予約されている、MemRpReservedUnit のメモリー容量
ldomMemRpCapacityUnit および ldomMemRpReservedUnit	整数	読み取り専用	次のメモリー割り当て単位のいずれか 1 つです。 ■ 1 は KB ■ 2 は MB ■ 3 は GB ■ 4 は バイト 指定されていない場合、単位の値はバイトです。

表 21-7 暗号化リソースプールのスカラー変数

名前	データ型	アクセス	説明
ldomCryptoRpCapacity	整数	読み取り専用	リソースプールが許可する最大予約数
ldomCryptoRpReserved	整数	読み取り専用	リソースプールから現在予約されている暗号化装置の数

表 21-8 I/O バスリソースプールのスカラー変数

名前	データ型	アクセス	説明
ldomIOBusRpCapacity	整数	読み取り専用	プールが許可する最大予約数
ldomIOBusRpReserved	整数	読み取り専用	リソースプールから現在予約されている I/O バスの数

表 21-9 CMI リソースプールのスカラー変数

名前	データ型	アクセス	説明
ldomCMIRpCapacity	整数	読み取り専用	プールが許可する最大予約数
ldomCMIRpReserved	整数	読み取り専用	リソースプールから現在予約されている CMI リソースの数

仮想 CPU テーブル (ldomVcpuTable)

ldomVcpuTable は、すべてのドメインが使用する仮想 CPU を示します。

表 21-10 仮想 CPU テーブル (ldomVcpuTable)

名前	データ型	アクセス	説明
ldomVcpuLdomIndex	整数	読み取り専用	仮想 CPU を含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVcpuIndex	整数	アクセス不可	このテーブルの仮想 CPU にインデックスを作成するために使用される整数
ldomVcpuDeviceID	表示文字列	読み取り専用	仮想 CPU の識別子 (VID)
ldomVcpuOperationalStatus	整数	読み取り専用	次の CPU ステータスのいずれか 1 つです。 1 は Unknown 2 は Other 3 は OK 4 は Degraded 5 は Stressed 6 は Predictive failure 7 は Error 8 は Nonrecoverable error 9 は Starting 10 は Stopping 11 は Stopped 12 は In service

名前	データ型	アクセス	説明
			13 は No contact
			14 は Lost communication
			15 は Aborted
			16 は Dormant
			17 は Supporting entity in error
			18 は Completed
			19 は Power mode
			Logical Domains Manager は CPU 状態を提供しないので、デフォルト値は 1 (Unknown) です。
ldomVcpuPhysBind	表示文字列	読み取り専用	物理バインディング (PID)。この仮想 CPU に割り当てられているハードウェアスレッド (ストランド) の識別子を含んでいます。この識別子はコアとチップを一意に識別します。
ldomVcpuPhysBindUsage	整数	読み取り専用	この仮想 CPU によって使用されているスレッドの全容量を MHz で示します。たとえば、あるスレッドが最大 1 GHz で動作できるとします。その容量の半分 (スレッドの 50%) だけがこの仮想 CPU に割り当てられている場合、プロパティ値は 500 になります。
ldomVcpuCoreID	表示文字列	読み取り専用	コアの識別子 (コア ID)。
ldomVcpuUtilPercent	表示文字列	読み取り専用	仮想 CPU の使用率を示します。

仮想メモリーテーブル

ドメインのメモリー空間は、*実メモリー*として参照されます。つまり、これは*仮想メモリー*です。ハイパーバイザで検出されるホストプラットフォームのメモリー空間は、*物理メモリー*として参照されます。ハイパーバイザは、物理メモリーのブロックを割り当てて、ドメインで使用される実メモリーのブロックを形成します。

次の例では、要求された記憶域サイズが、単一の大きなメモリーブロックに割り当てられる代わりに、2 つのメモリーブロックに分けられることを示しています。たとえば、ドメインが 521M バイトの実メモリーを要求しているとします。メモリーには、`{physical-address, real-address, size}`

形式を使用して、ホストシステム上の 256M バイトの 2 つのブロックを物理メモリーとして割り当てられます。

```
{0x1000000, 0x1000000, 256}, {0x2000000, 0x2000000, 256}
```

ドメインには、ゲストドメインに割り当てられる物理メモリーセグメントを 64 個まで構成できます。したがって、各メモリーセグメントを保持するには、表示文字列ではなく、補助テーブルが使用されます。表示文字列には、255 文字の制限があります。

仮想メモリーテーブル (ldomVmemTable)

ldomVmemTable は、ドメインが使用する仮想メモリーのプロパティを示します。

表 21-11 仮想メモリーテーブル (ldomVmemTable)

名前	データ型	アクセス	説明
ldomVmemLdomIndex	整数	読み取り専用	仮想メモリーを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVmemIndex	整数	アクセス不可	このテーブルの仮想メモリーにインデックスを作成するために使用される整数
ldomVmemNumberOfBlocks	整数	読み取り専用	仮想メモリーのブロック数

仮想メモリーの物理バインディングテーブル (ldomVmemPhysBindTable)

ldomVmemPhysBindTable は、すべてのドメインの物理メモリーセグメントを含む補助テーブルです。

表 21-12 仮想メモリーの物理バインディングテーブル (ldomVmemPhysBindTable)

名前	データ型	アクセス	説明
ldomVmemPhysBindLdomIndex	整数	読み取り専用	物理メモリーセグメントを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVmemPhysBind	表示文字列	読み取り専用	この仮想メモリーブロックに次の形式で割り当てられる物理メモリーのリスト: { <i>physical-address</i> , <i>real-address</i> , <i>size</i> }

仮想ディスクテーブル

仮想ディスクサービス (vds) とこれに割り当てられた物理デバイス (vdsdev) によって、仮想ディスク機能が Oracle VM Server for SPARC テクノロジーで実現できます。仮想ディスクサービスは、いくつかのローカルボリューム (物理ディスクまたはファイルシステム) をエクスポートします。仮想ディスクサービスが指定された場合、次のものが含まれます。

- 補助デバイス (vdsdev) の完全な /dev パス
- サービスに追加されるデバイスの一意の名前 (ボリューム名)

1 つのディスクサービスには、1 つ以上のディスク、ディスクスライス、およびファイルシステムをバインドできます。各ディスクには、一意の名前とボリューム名があります。このボリューム名は、ディスクがサービスにバインドされるときに使用されます。Logical Domains Manager は、仮想ディスクサービスとその論理ボリュームから仮想ディスククライアント (vdisk) を作成します。

仮想ディスクサービステーブル (ldomVdsTable)

ldomVdsTable は、すべてのドメインの仮想ディスクサービスを示します。

表 21-13 仮想ディスクサービステーブル (ldomVdsTable)

名前	データ型	アクセス	説明
ldomVdsLdomIndex	整数	読み取り専用	仮想ディスクサービスを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVdsIndex	整数	アクセス不可	このテーブルの仮想ディスクサービスにインデックスを作成するために使用される整数
ldomVdsServiceName	表示文字列	読み取り専用	仮想ディスクサービスのサービス名。プロパティ値は、ldm add-vds コマンドによって指定される service-name です。
ldomVdsNumofAvailVolume	整数	読み取り専用	この仮想ディスクサービスによってエクスポートされる論理ボリュームの数
ldomVdsNumofUsedVolume	整数	読み取り専用	この仮想ディスクサービスに使用 (バインド) される論理ボリュームの数

仮想ディスクサービスデバイステーブル (ldomVdsdevTable)

ldomVdsdevTable は、すべての仮想ディスクサービスが使用する仮想ディスクサービスデバイスを示します。

表 21-14 仮想ディスクサービスデバイステーブル (ldomVdsdevTable)

名前	データ型	アクセス	説明
ldomVdsdevVdsIndex	整数	読み取り専用	仮想ディスクデバイスを含む仮想ディスクサービスを表す ldomVdsTable にインデックスを作成するために使用される整数
ldomVdsdevIndex	整数	アクセス不可	このテーブルの仮想ディスクサービスデバイスにインデックスを作成するために使用される整数
ldomVdsdevVolumeName	表示文字列	読み取り専用	仮想ディスクサービスデバイスのボリューム名。このプロパティは、仮想ディスクサービスに追加されるデバイスの一意の名前を指定します。この名前は、デバイスを追加するために、仮想ディスクサービスによってクライアントにエクスポートされます。プロパティ値は、ldm add-vdsdev コマンドによって指定される <i>volume-name</i> です。
ldomVdsdevDevPath	表示文字列	読み取り専用	物理ディスクデバイスのパス名。プロパティ値は、ldm add-vdsdev コマンドによって指定される <i>backend</i> です。
ldomVdsdevOptions	表示文字列	読み取り専用	ディスクデバイスの 1 つ以上のオプション。ro、slice、または excl
ldomVdsdevMPGroup	表示文字列	読み取り専用	ディスクデバイスのマルチパスグループ名

仮想ディスクテーブル (ldomVdiskTable)

ldomVdiskTable は、すべてのドメインの仮想ディスクを示します。

表 21-15 仮想ディスクテーブル (ldomVdiskTable)

名前	データ型	アクセス	説明
ldomVdiskLdomIndex	整数	読み取り専用	仮想ディスクデバイスを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVdiskVdsDevIndex	整数	読み取り専用	仮想ディスクサービスデバイスを表す ldomVdsdevTable にインデックスを作成するために使用される整数
ldomVdiskIndex	整数	アクセス不可	このテーブルの仮想ディスクにインデックスを作成するために使用される整数

名前	データ型	アクセス	説明
ldomVdiskName	表示文字列	読み取り専用	仮想ディスクの名前。プロパティ値は、ldm add-vdisk コマンドによって指定される <i>disk-name</i> です。
ldomVdiskTimeout	整数	読み取り専用	仮想ディスククライアントと仮想ディスクサーバーの間に接続を確立するためのタイムアウト (秒単位)
ldomVdiskID	表示文字列	読み取り専用	仮想ディスクの識別子

次の図は、仮想ディスクテーブルとドメインテーブルの関係性を定義するために、インデックスがどのように使用されるかを示しています。インデックスは、次のように使用されます。

- ldomVdsTable および ldomVdiskTable の ldomIndex は ldomTable を指します。
- ldomVdsdevTable の VdsIndex は ldomVdsTable を指します。
- ldomVdiskTable の VdsDevIndex は ldomVdsdevTable を指します。

図 21-3 仮想ディスクテーブルとドメインテーブルとの関係



仮想ネットワークテーブル

Oracle VM Server for SPARC 仮想ネットワークサポートにより、ゲストドメインは物理 Ethernet デバイスを介して相互に通信したり、外部ホストと通信したりすることができます。仮想ネットワークには、次の主要コンポーネントが含まれています。

- 仮想スイッチ (vsw)
- 仮想ネットワークデバイス (vnet)

サービスドメイン上に仮想スイッチを作成したあとで、物理ネットワークデバイスを仮想スイッチにバインドできます。その後、通信に仮想スイッチサービスを使用するドメイン向けに、仮想ネットワークデバイスを作成できます。仮想スイッチサービスは、同じ仮想スイッチに接続することで、ほかのドメインと通信します。物理デバイスが仮想スイッチにバインドされている場合、仮想スイッチサービスは外部ホストと通信します。

仮想スイッチサービステーブル (ldomVswTable)

ldomVswTable は、すべてのドメインの仮想スイッチサービスを示します。

表 21-16 仮想スイッチサービステーブル (ldomVswTable)

名前	データ型	アクセス	説明
ldomVswLdomIndex	整数	読み取り専用	仮想スイッチサービスを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVswIndex	整数	アクセス不可	このテーブルの仮想スイッチデバイスにインデックスを作成するために使用される整数
ldomVswServiceName	表示文字列	読み取り専用	仮想スイッチサービス名
ldomVswMacAddress	表示文字列	読み取り専用	仮想スイッチが使用する MAC アドレス
ldomVswPhysDevPath	表示文字列	読み取り専用	仮想ネットワークスイッチの物理デバイスパス。物理デバイスが仮想スイッチにバインドされていない場合、プロパティ値は null です。
ldomVswMode	表示文字列	読み取り専用	実行中のクラスタノードの値は mode=sc
ldomVswDefaultVlanID	表示文字列	読み取り専用	仮想スイッチのデフォルトの VLAN ID
ldomVswPortVlanID	表示文字列	読み取り専用	仮想スイッチのポート VLAN ID
ldomVswVlanID	表示文字列	読み取り専用	仮想スイッチの VLAN ID
ldomVswLinkprop	表示文字列	読み取り専用	物理ネットワークデバイスに基づいてリンクステータスを報告するための値は linkprop=phys-state
ldomVswMtu	整数	読み取り専用	仮想スイッチデバイスの最大転送単位 (MTU)
ldomVswID	表示文字列	読み取り専用	仮想スイッチデバイスの識別子
ldomVswInterVnetLink	表示文字列	読み取り専用	inter-vnet 通信の LDC チャネル割り当ての状態。値は on または off です。

仮想ネットワークデバイステーブル (ldomVnetTable)

ldomVnetTable は、すべてのドメインの仮想ネットワークデバイスを示します。

表 21-17 仮想ネットワークデバイステーブル (ldomVnetTable)

名前	データ型	アクセス	説明
ldomVnetLdomIndex	整数	読み取り専用	仮想ネットワークデバイスを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVnetVswIndex	整数	読み取り専用	仮想スイッチサービステーブルにインデックスを作成するために使用される整数
ldomVnetIndex	整数	アクセス不可	このテーブルの仮想ネットワークデバイスにインデックスを作成するために使用される整数
ldomVnetDevName	表示文字列	読み取り専用	仮想ネットワークデバイス名。プロパティ値は、ldm add-vnet コマンドによって指定される net-dev プロパティです。
ldomVnetDevMacAddress	表示文字列	読み取り専用	このネットワークデバイスの MAC アドレス。プロパティ値は、ldm add-vnet コマンドによって指定される mac-addr プロパティです。
ldomVnetMode	表示文字列	読み取り専用	仮想ネットワークデバイスで NIU ハイブリッド I/O を使用するための値は mode=hybrid
ldomVnetPortVlanID	表示文字列	読み取り専用	仮想ネットワークデバイスのポート VLAN ID
ldomVnetVlanID	表示文字列	読み取り専用	仮想ネットワークデバイスの VLAN ID
ldomVnetLinkprop	表示文字列	読み取り専用	物理ネットワークデバイスに基づいてリンクステータスを報告するための値は linkprop=phys-state
ldomVnetMtu	整数	読み取り専用	仮想ネットワークデバイスの MTU
ldomVnetID	表示文字列	読み取り専用	仮想ネットワークデバイスの識別子

仮想コンソールテーブル

Oracle VM Server for SPARC サービスドメインは、仮想ネットワーク端末サービス (vNTS) を提供します。vNTS は、仮想コンソール端末集配信装置 (vcc) と呼ばれる仮想コンソールサービスを幅広いポート番号とともに提供します。それぞれの仮想コンソール端末集配信装置は複数のコンソールグループ (vcons) を持ち、各グループにはポート番号が割り当てられています。各グループには複数のドメインを含めることができます。

仮想コンソール端末集配信装置テーブル (ldomVccTable)

ldomVccTable は、すべてのドメインの仮想コンソール端末集配信装置を示します。

表 21-18 仮想コンソール端末集配信装置テーブル (ldomVccTable)

名前	データ型	アクセス	説明
ldomVccLdomIndex	整数	読み取り専用	仮想コンソールサービスを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomVccIndex	整数	アクセス不可	このテーブルの仮想コンソール端末集配信装置にインデックスを作成するために使用される整数
ldomVccName	表示文字列	読み取り専用	仮想コンソール端末集配信装置の名前。プロパティ値は、ldm add-vcc コマンドによって指定される vcc-name です。
ldomVccPortRangeLow	整数	読み取り専用	仮想コンソール端末集配信装置によって使用される TCP ポートの範囲の小さい番号。プロパティ値は、ldm add-vcc コマンドによって指定される port-range の x 部分です。
ldomVccPortRangeHigh	整数	読み取り専用	仮想コンソール端末集配信装置によって使用される TCP ポートの範囲の大きい番号。プロパティ値は、ldm add-vcc コマンドによって指定される port-range の y 部分です。

仮想コンソールグループテーブル (ldomVconsTable)

ldomVconsTable は、すべての仮想コンソールサービスの仮想コンソールグループを示します。この表には、各ドメインでコンソールロギングが有効であるか無効であるかも示します。

表 21-19 仮想コンソールグループテーブル (ldomVconsTable)

名前	データ型	アクセス	説明
ldomVconsIndex	整数	アクセス不可	このテーブルの仮想グループにインデックスを作成するために使用される整数
ldomVconsGroupName	表示文字列	読み取り専用	仮想コンソールを接続するグループ名。プロパティ値は、ldm set-vcons コマンドによって指定される group です。

名前	データ型	アクセス	説明
ldomVconsLog	表示文字列	読み取り専用	コンソールロギングのステータス。プロパティ値は、ldm set-vcons コマンドによって指定される文字列 on または off です。 グループに複数のドメインが含まれている場合、このプロパティは、ldm set-vcons コマンドによって変更された直近のドメインのコンソールロギングのステータスを示します。
ldomVconsPortNumber	整数	読み取り専用	このグループに割り当てられるポート番号。プロパティ値は、ldm set-vcons コマンドによって指定される port です。

仮想コンソール関係テーブル (ldomVconsVccRelTable)

ldomVconsVccRelTable には、ドメイン、仮想コンソール端末集配信装置、およびコンソールグループのテーブル間の関係を示すインデックス値が含まれています。

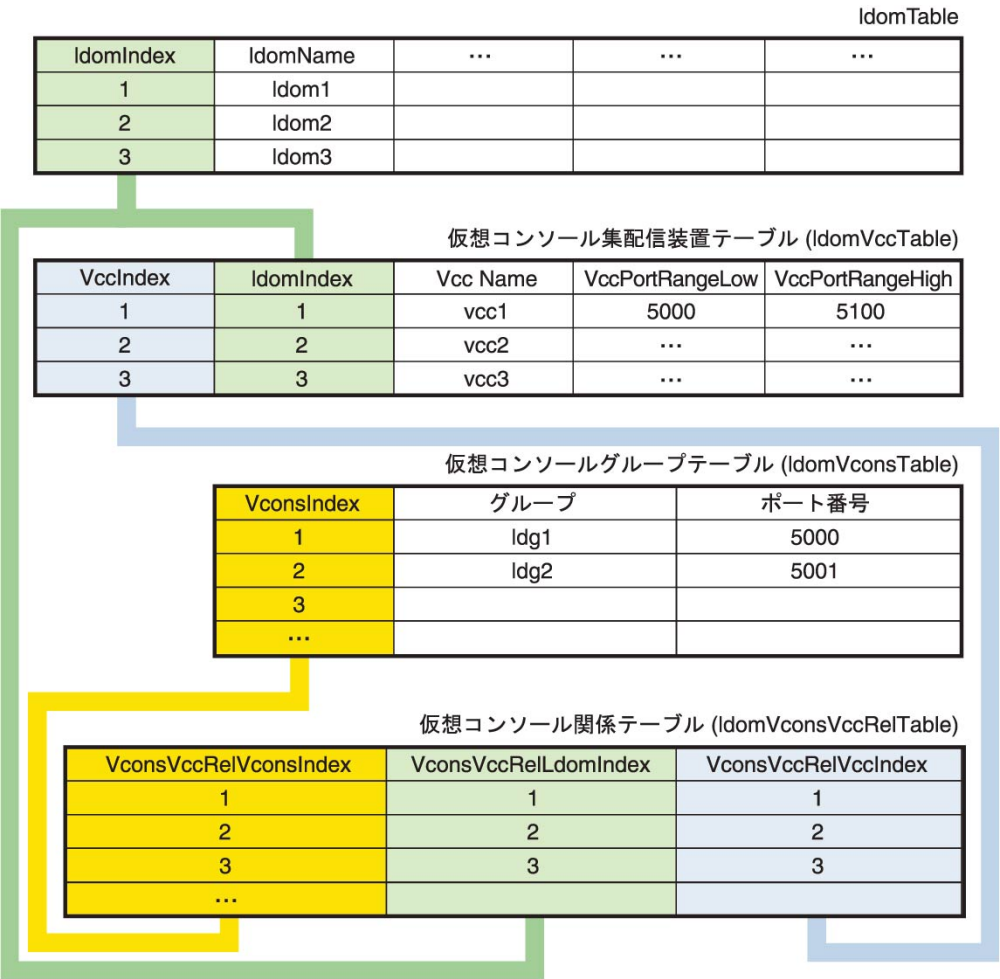
表 21-20 仮想コンソール関係テーブル (ldomVconsVccRelTable)

名前	データ型	アクセス	説明
ldomVconsVccRelVconsIndex	整数	読み取り専用	ldomVconsTable の ldomVconsIndex の値
ldomVconsVccRelLdomIndex	整数	読み取り専用	ldomTable の ldomIndex の値
ldomVconsVccRelVccIndex	整数	読み取り専用	ldomVccTable の ldomVccIndex の値

次の図は、仮想コンソールテーブルとドメインテーブルの間の関係を定義するために、インデックスがどのように使用されるかを示しています。インデックスは、次のように使用されます。

- ldomVccTable および ldomVconsVccRelTable の ldomIndex は ldomTable を指します。
- ldomVconsVccRelTable の VccIndex は ldomVccTable を指します。
- ldomVconsVccRelTable の VconsIndex は ldomVconsTable を指します。

図 21-4 仮想コンソールテーブルとドメインテーブルとの関係



暗号化装置テーブル (ldomCryptoTable)

ldomCryptoTable は、すべてのドメインが使用する暗号化装置を示します。暗号化装置は、モジュー演算ユニット (MAU) と呼ばれる場合があります。

表 21-21 暗号化装置テーブル (ldomCryptoTable)

名前	データ型	アクセス	説明
ldomCryptoLdomIndex	整数	読み取り専用	暗号化装置を含むドメインを表す ldomTable へのインデックスとして使用される整数装置
ldomCryptoIndex	整数	アクセス不可	このテーブルの暗号化装置にインデックスを作成するために使用される整数
ldomCryptoCpuSet	表示文字列	読み取り専用	MAU-unit cpuset に割り当てられる CPU のリスト。例: {0, 1, 2, 3}

I/O バステーブル (ldomIOBusTable)

ldomIOBusTable は、すべてのドメインが使用する物理 I/O デバイスと PCI バスを示します。

表 21-22 I/O バステーブル (ldomIOBusTable)

名前	データ型	アクセス	説明
ldomIOBusLdomIndex	整数	読み取り専用	I/O バスを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomIOBusIndex	整数	アクセス不可	このテーブルの I/O バスにインデックスを作成するために使用される整数
ldomIOBusName	表示文字列	読み取り専用	物理 I/O デバイス名
ldomIOBusPath	表示文字列	読み取り専用	物理 I/O デバイスパス
ldomIOBusOptions	表示文字列	読み取り専用	物理 I/O デバイスオプション

CMI テーブル (ldomCMITable)

ldomCMITable は、すべてのドメインの CMI リソース情報を示します。

表 21-23 CMI テーブル (ldomCMITable)

名前	データ型	アクセス	説明
ldomCMILdomIndex	整数	読み取り専用	CMI リソースを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomCMIIndex	整数	アクセス不可	このテーブル内の CMI リソースにインデックスを作成するために使用される整数
ldomCMIID	表示文字列	読み取り専用	CMI リソースの識別子 (CMI ID)
ldomCMICpuSet	表示文字列	読み取り専用	CMI リソースにマップされる CPU のリスト

名前	データ型	アクセス	説明
ldomCMICores	表示文字列	読み取り専用	CMI リソースにマップされるコアのリスト

コアテーブル (ldomCoreTable)

ldomCoreTable は、すべてのドメインのコア情報 (core-id、cpuset など) を示します。

表 21-24 コアテーブル (ldomCoreTable)

名前	データ型	アクセス	説明
ldomCoreLdomIndex	整数	読み取り専用	コアを含むドメインを表す ldomTable へのインデックスとして使用される整数
ldomCoreIndex	整数	アクセス不可	このテーブルのコアにインデックスを作成するために使用される整数
ldomCoreID	表示文字列	読み取り専用	コアの識別子 (コア ID)
ldomCoreCpuSet	表示文字列	読み取り専用	コア cpuset に割り当てられる CPU のリスト

ドメインバージョン情報のスカラー変数

Logical Domains Manager プロトコルは、メジャー番号とマイナー番号から構成されるドメインバージョンをサポートしています。Oracle VM Server for SPARC MIB には、ドメインバージョン情報を示すスカラー変数が設定されています。

表 21-25 ドメインバージョン情報のスカラー変数

名前	データ型	アクセス	説明
ldomVersionMajor	整数	読み取り専用	メジャーバージョン番号
ldomVersionMinor	整数	読み取り専用	マイナーバージョン番号

ldomVersionMajor と ldomVersionMinor の値は、ldm list -p コマンドによって表示されるバージョンと同等です。たとえば、次のように表示されます。

```
$ ldm ls -p
VERSION 1.6
...
```

```
$ snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMajor.0
SUN-LDOM-MIB::ldomVersionMajor.0 = INTEGER: 1
```

```
$ snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomVersionMinor.0
SUN-LDOM-MIB::ldomVersionMinor.0 = INTEGER: 5
```

SNMPトラップの使用

このセクションでは、トラップを送受信するためのシステムの設定方法について説明します。また、論理ドメイン (ドメイン) の変更通知を受信するために使用するトラップのほか、使用可能なほかのトラップについても説明します。

snmptrapd デーモンは、すべての受信トラップを自動的に受け入れるわけではありません。その代わりに、承認された SNMP v1 と v2c コミュニティ文字列、または SNMPv3 ユーザー (あるいはその両方) を使用してデーモンを構成する必要があります。承認されていないトラップまたは通知は破棄されます。snmptrapd.conf(4) または snmptrapd.conf(5) のマニュアルページを参照してください。

Oracle VM Server for SPARC MIB モジュールトラップの使用

受信通知にはアクセス制御チェックが適用されます。snmptrapd を適切な構成ファイルを指定せず、または同等のアクセス制御設定を指定して実行した場合、このようなトラップは処理されません。snmptrapd.conf(4) または snmptrapd.conf(5) のマニュアルページを参照してください。

▼ トラップの送信方法

1. トラップを構成します。

/etc/net-snmp/snmp/snmpd.conf SNMP 構成ファイルを編集して、トラップ、通知バージョン、および送信先を定義する指令を追加します。

/etc/net-snmp/snmp/snmpd.conf ファイルを編集するには pfedit コマンドを使用する必要があります。

```
trapcommunity string --> define community string to be used when sending traps
trapsink host[community [port]] --> to send v1 traps
trap2sink host[community [port]] --> to send v2c traps
informsink host[community [port]] --> to send informs
```

詳細は、snmpd.conf(4) または snmpd.conf(5) のマニュアルページを参照してください。

たとえば、次の指令では、トラップの送信時にコミュニティ文字列として public 文字列が使用され、v1 トラップが localhost 送信先に送信されることを示します。

```
trapcommunity public
trapsink localhost
```

2. **/usr/etc/snmp/snmptrapd.conf SNMP trapd 構成ファイルを作成または編集して、アクセス制御設定を構成します。**

/etc/net-snmp/snmp/snmpd.conf ファイルを編集するには pfedit コマンドを使用する必要があります。

次の例では、トラップの送信を承認されているユーザー (public)、および受信トラップの処理方法 (log,execute,net) を示します。snmptrapd.conf(4) または snmptrapd.conf(5) のマニュアルページを参照してください。

```
authCommunity log,execute,net public
```

3. **SNMP トラップメッセージを受信するには、SNMP トラップデーモンユーティリティ snmptrapd を起動します。**

例 21-4 SNMP v1 および v2c トラップの送信

この例では、v1 トラップおよび v2c トラップの両方を、同じホスト上で動作する SNMP トラップデーモンに送信しています。/etc/net-snmp/snmp/snmpd.conf ファイルを次のディレクティブで更新します。

```
trapcommunity public
trapsink localhost
trap2sink localhost
```

▼ トラップの受信方法

- **SNMP トラップデーモンユーティリティを起動します。**

出力形式オプションについては、snmptrapd(1M) のマニュアルページを参照してください。

snmptrapd ユーティリティは、SNMP TRAP メッセージを受信してログに記録する SNMP アプリケーションです。たとえば、次の snmptrapd コマンドは、新しいドメインが ldg2 (ldomName = ldg2) という名前で作成された (ldomTrapDesc = Ldom Created) ことを示しています。

```
# /usr/sbin/snmptrapd -f -Le -F \
```

```
"TRAP from %B on %m/%L/%y at %h:%j:%k Enterprise=%N Type=%w SubType=%q\n
with Varbinds: %v\nSecurity info:%P\n\n" localhost:162
NET-SNMP version 5.4.1
TRAP from localhost on 6/27/2012 at 12:13:48
Enterprise=SUN-LDOM-MIB::ldomMIBTraps Type=6 SubType=SUN-LDOM-MIB::ldomCreate
with Varbinds: SUN-LDOM-MIB::ldomIndexNotif = INTEGER: 3
SUN-LDOM-MIB::ldomName = STRING: ldg2 SUN-LDOM-MIB::ldomTrapDesc = STRING:
Ldom Created
Security info:TRAP, SNMP v1, community public
```

読みやすくするために、-F オプション引数文字列を 2 行に分けています。

Oracle VM Server for SPARC MIB トラップの説明

このセクションでは、使用可能な Oracle VM Server for SPARC MIB トラップについて説明します。

ドメインの作成 (ldomCreate)

このトラップは、ドメインが作成されたときに通知を行います。

表 21-26 ドメインの作成トラップ (ldomCreate)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	ドメインの名前
ldomTrapDesc	表示文字列	トラップの説明

ドメインの削除 (ldomDestroy)

このトラップは、ドメインが削除されたときに通知を行います。

表 21-27 ドメインの削除トラップ (ldomDestroy)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	ドメインの名前
ldomTrapDesc	表示文字列	トラップの説明

ドメインの状態変更 (ldomStateChange)

このトラップは、ドメインの動作状態の変更を通知します。

表 21-28 ドメインの状態変更トラップ (ldomStateChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	ドメインの名前
ldomOperState	整数	ドメインの新しい状態
ldomStatePrev	整数	ドメインの前の状態
ldomTrapDesc	表示文字列	トラップの説明

仮想 CPU の変更 (ldomVCpuChange)

このトラップは、ドメイン中の仮想 CPU の数が増えたり減ったりしたときに通知を行います。

表 21-29 ドメインの仮想 CPU の変更トラップ (ldomVCpuChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想 CPU を含むドメインの名前
ldomNumVCPU	整数	ドメインの仮想 CPU の新しい数
ldomNumVCPUPrev	整数	ドメインの仮想 CPU の前の数
ldomTrapDesc	表示文字列	トラップの説明

仮想メモリーの変更 (ldomVMemChange)

このトラップは、ドメイン中の仮想メモリーの容量が増えたり減ったりしたときに通知を行います。

表 21-30 ドメインの仮想メモリーの変更トラップ (ldomVMemChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想メモリーを含むドメインの名前

名前	データ型	説明
ldomMemSize	整数	ドメインの仮想メモリーの容量
ldomMemSizePrev	整数	ドメインの仮想メモリーの前の容量
ldomMemUnit	整数	仮想メモリーのメモリー単位。次のいずれか 1 つです。 ■ 1 は KB ■ 2 は MB ■ 3 は GB ■ 4 は バイト 指定されていない場合、単位の値はバイトです。
ldomMemUnitPrev	整数	前の仮想メモリーのメモリー単位。次のいずれか 1 つです。 ■ 1 は KB ■ 2 は MB ■ 3 は GB ■ 4 は バイト 指定されていない場合、単位の値はバイトです。
ldomTrapDesc	表示文字列	トラップの説明

仮想ディスクサービスの変更 (ldomVdsChange)

このトラップは、ドメインの仮想ディスクサービスが変更されたときに通知を行います。

表 21-31 ドメインの仮想ディスクサービスの変更トラップ (ldomVdsChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想ディスクサービスを含むドメインの名前
ldomVdsServiceName	表示文字列	変更された仮想ディスクサービスの名前
ldomChangeFlag	整数	仮想ディスクサービスに行われた次の変更のいずれか 1 つを示します。 ■ 1 は追加 ■ 2 は変更 ■ 3 は削除

名前	データ型	説明
ldomTrapDesc	表示文字列	トラップの説明

仮想ディスクの変更 (ldomVdiskChange)

このトラップは、ドメインの仮想ディスクが変更されたときに通知を行います。

表 21-32 仮想ディスクの変更トラップ (ldomVdiskChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想ディスクデバイスを含むドメインの名前
ldomVdiskName	表示文字列	変更された仮想ディスクデバイスの名前
ldomChangeFlag	整数	仮想ディスクサービスに行われた次の変更のいずれか 1 つを示します。 ■ 1 は追加 ■ 2 は変更 ■ 3 は削除
ldomTrapDesc	表示文字列	トラップの説明

仮想スイッチの変更 (ldomVswChange)

このトラップは、ドメインの仮想スイッチが変更されたときに通知を行います。

表 21-33 仮想スイッチの変更トラップ (ldomVswChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想スイッチサービスを含むドメインの名前
ldomVswServiceName	表示文字列	変更された仮想スイッチサービスの名前
ldomChangeFlag	整数	仮想スイッチサービスに行われた次の変更のいずれか 1 つを示します。 ■ 1 は追加 ■ 2 は変更 ■ 3 は削除
ldomTrapDesc	表示文字列	トラップの説明

仮想ネットワークの変更 (ldomVnetChange)

このトラップは、ドメインの仮想ネットワークが変更されたときに通知を行います。

表 21-34 仮想ネットワークの変更トラップ (ldomVnetChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想ネットワークデバイスを含むドメインの名前
ldomVnetDevName	表示文字列	ドメインの仮想ネットワークデバイスの名前
ldomChangeFlag	整数	仮想ディスクサービスに行われた次の変更のいずれか 1 つを示します。 ■ 1 は追加 ■ 2 は変更 ■ 3 は削除
ldomTrapDesc	表示文字列	トラップの説明

仮想コンソール端末集配信装置の変更 (ldomVccChange)

このトラップは、ドメインの仮想コンソール端末集配信装置が変更されたときに通知を行います。

表 21-35 仮想コンソール端末集配信装置の変更トラップ (ldomVccChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想コンソール端末集配信装置を含むドメインの名前
ldomVccName	表示文字列	変更された仮想コンソール端末集配信装置サービスの名前
ldomChangeFlag	整数	仮想コンソール端末集配信装置に行われた次の変更のいずれか 1 つを示します。 ■ 1 は追加 ■ 2 は変更 ■ 3 は削除
ldomTrapDesc	表示文字列	トラップの説明

仮想コンソールグループの変更 (ldomVconsChange)

このトラップは、ドメインの仮想コンソールグループが変更されたときに通知を行います。

表 21-36 仮想コンソールグループの変更トラップ (ldomVconsChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	仮想コンソールグループを含むドメインの名前
ldomVconsGroupName	表示文字列	変更された仮想コンソールグループの名前
ldomChangeFlag	整数	仮想コンソールグループに行われた次の変更のいずれか 1 つを示します。 ■ 1 は追加 ■ 2 は変更 ■ 3 は削除
ldomTrapDesc	表示文字列	トラップの説明

CMI リソースの変更 (ldomCMIChange)

このトラップは、ドメイン内の CMI リソースの数が増えたり減ったりしたときに通知されます。

表 21-37 CMI リソースの変更トラップ (ldomCMIChange)

名前	データ型	説明
ldomIndexNotif	整数	ldomTable へのインデックス
ldomName	表示文字列	CMI リソースを含むドメインの名前
ldomNumCMI	整数	ドメインの CMI リソースの新しい数
ldomNumCMIPrev	整数	ドメインの CMI リソースの以前の数
ldomTrapDesc	表示文字列	トラップの説明

ドメインの起動と停止

このセクションでは、ドメインの起動と停止に使用するアクティブな管理操作について説明します。ドメインテーブル ldomTable の ldomAdminState プロパティの値を設定することによって、

これらのアクティブな管理操作を制御できます。[表21-1「ドメインテーブル \(ldomTable\)」](#)を参照してください。

▼ ドメインの起動方法

この手順では、既存のバインドされたドメインを起動する方法について説明します。指定されたドメイン名を持つドメインが存在しない場合、またはまだバインドされていない場合、この操作は失敗します。

1. *domain-name* ドメインが存在し、すでにバインドされていることを確認します。

```
# ldm list domain-name
```

2. **ldomTable** の *domain-name* を特定します。

```
# snmpwalk -v1 -c public localhost SUN-LDOM-MIB::ldomTable
SUN-LDOM-MIB::ldomName.1 = STRING: primary
SUN-LDOM-MIB::ldomName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldomAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldomAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldomOperState.2 = INTEGER: bound(6)
SUN-LDOM-MIB::ldomNumVCpu.1 = INTEGER: 8
SUN-LDOM-MIB::ldomNumVCpu.2 = INTEGER: 4
SUN-LDOM-MIB::ldomMemSize.1 = INTEGER: 3360
SUN-LDOM-MIB::ldomMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldomMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomNumCrypto.1 = INTEGER: 1
SUN-LDOM-MIB::ldomNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldomNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldomNumIOBus.2 = INTEGER: 0
SUN-LDOM-MIB::ldomUUID.1 = STRING: 5f8817d4-5d2e-6f7d-c4af-91b5b34b5723
SUN-LDOM-MIB::ldomUUID.2 = STRING: 11284146-87ca-4877-8d80-cd0f60d5ec26
SUN-LDOM-MIB::ldomMacAddress.1 = STRING: 00:14:4f:46:47:d6
SUN-LDOM-MIB::ldomMacAddress.2 = STRING: 00:14:4f:f8:d5:6c
SUN-LDOM-MIB::ldomHostID.1 = STRING: 0x844647d6
SUN-LDOM-MIB::ldomHostID.2 = STRING: 0x84f8d56c
SUN-LDOM-MIB::ldomFailurePolicy.1 = STRING: ignore
SUN-LDOM-MIB::ldomFailurePolicy.2 = STRING: ignore
SUN-LDOM-MIB::ldomMaster.1 = STRING:
SUN-LDOM-MIB::ldomMaster.2 = STRING:
SUN-LDOM-MIB::ldomExtMapinSpace.1 = STRING: off
SUN-LDOM-MIB::ldomExtMapinSpace.2 = STRING: off
SUN-LDOM-MIB::ldomWholeCore.1 = INTEGER: 0
SUN-LDOM-MIB::ldomWholeCore.2 = INTEGER: 0
SUN-LDOM-MIB::ldomCpuArch.1 = STRING: native
SUN-LDOM-MIB::ldomCpuArch.2 = STRING: native
```

```
SUN-LDOM-MIB::ldomShutdownGroup.1 = INTEGER: 0
SUN-LDOM-MIB::ldomShutdownGroup.2 = INTEGER: 15
SUN-LDOM-MIB::ldomPerfCounters.1 = STRING: htstrand
SUN-LDOM-MIB::ldomPerfCounters.2 = STRING: global,htstrand
SUN-LDOM-MIB::ldomNumCMI.1 = INTEGER: 0
SUN-LDOM-MIB::ldomNumCMI.2 = INTEGER: 0
```

3. *domain-name* ドメインを起動します。

snmpset コマンドを使用して、ldomAdminState プロパティに値 1 を設定することで、ドメインを起動します。*n* には起動するドメインを指定します。

```
# snmpset -v version -c community-string hostname \
SUN-LDOM-MIB::ldomTable.1.ldomAdminState.n = 1
```

4. 次のいずれかのコマンドを使用して、*domain-name* ドメインがアクティブであることを確認します。

```
■ # ldm list domain-name
■ # snmpget -v version -c community-string hostname SUN-LDOM-MIB::ldomOperState.n
```

例 21-5 ゲストドメインの起動

この例では、ldomAdminState プロパティを 1 に設定する前に、LdomMibTest_1 ドメインが存在してすでにバインドされていることを確認しています。最後に、ldm list LdomMibTest_1 コマンドを使用して、LdomMibTest_1 ドメインがアクティブであることを確認しています。

```
# ldm list LdomMibTest_1
# snmpset -v1 -c private localhost SUN-LDOM-MIB::ldomTable.1.ldomAdminState.2 = 1
# ldm list LdomMibTest_1
```

ldm list コマンドを使用する代わりに、snmpget コマンドを使用して LdomMibTest_1 ドメインの状態を取得できます。

```
# snmpget -v1 -c public localhost SUN-LDOM-MIB::ldomOperState.2
```

ドメインがアクティブでない場合で snmpset を使用してドメインを起動するときは、ドメインをまずバインドしてから起動してください。

▼ ドメインの停止方法

この手順では、起動したドメインを停止する方法について説明します。ドメインでホストされているオペレーティングシステムインスタンスが停止します。

1. `ldomTable` の *domain-name* を特定します。

```
# snmpwalk -v1 -c public localhost SUN-LDOM-MIB::ldomTable
SUN-LDOM-MIB::ldomName.1 = STRING: primary
SUN-LDOM-MIB::ldomName.2 = STRING: LdomMibTest_1
SUN-LDOM-MIB::ldomAdminState.1 = INTEGER: 0
SUN-LDOM-MIB::ldomAdminState.2 = INTEGER: 0
SUN-LDOM-MIB::ldomOperState.1 = INTEGER: active(1)
SUN-LDOM-MIB::ldomOperState.2 = INTEGER: bound(6)
SUN-LDOM-MIB::ldomNumVCpu.1 = INTEGER: 8
SUN-LDOM-MIB::ldomNumVCpu.2 = INTEGER: 4
SUN-LDOM-MIB::ldomMemSize.1 = INTEGER: 3360
SUN-LDOM-MIB::ldomMemSize.2 = INTEGER: 256
SUN-LDOM-MIB::ldomMemUnit.1 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomMemUnit.2 = INTEGER: megabytes(2)
SUN-LDOM-MIB::ldomNumCrypto.1 = INTEGER: 1
SUN-LDOM-MIB::ldomNumCrypto.2 = INTEGER: 0
SUN-LDOM-MIB::ldomNumIOBus.1 = INTEGER: 2
SUN-LDOM-MIB::ldomNumIOBus.2 = INTEGER: 0
SUN-LDOM-MIB::ldomUUID.1 = STRING: 5f8817d4-5d2e-6f7d-c4af-91b5b34b5723
SUN-LDOM-MIB::ldomUUID.2 = STRING: 11284146-87ca-4877-8d80-cd0f60d5ec26
SUN-LDOM-MIB::ldomMacAddress.1 = STRING: 00:14:4f:46:47:d6
SUN-LDOM-MIB::ldomMacAddress.2 = STRING: 00:14:4f:f8:d5:6c
SUN-LDOM-MIB::ldomHostID.1 = STRING: 0x844647d6
SUN-LDOM-MIB::ldomHostID.2 = STRING: 0x84f8d56c
SUN-LDOM-MIB::ldomFailurePolicy.1 = STRING: ignore
SUN-LDOM-MIB::ldomFailurePolicy.2 = STRING: ignore
SUN-LDOM-MIB::ldomMaster.1 = STRING:
SUN-LDOM-MIB::ldomMaster.2 = STRING:
SUN-LDOM-MIB::ldomExtMapinSpace.1 = STRING: off
SUN-LDOM-MIB::ldomExtMapinSpace.2 = STRING: off
SUN-LDOM-MIB::ldomWholeCore.1 = INTEGER: 0
SUN-LDOM-MIB::ldomWholeCore.2 = INTEGER: 0
SUN-LDOM-MIB::ldomCpuArch.1 = STRING: native
SUN-LDOM-MIB::ldomCpuArch.2 = STRING: native
SUN-LDOM-MIB::ldomShutdownGroup.1 = INTEGER: 0
SUN-LDOM-MIB::ldomShutdownGroup.2 = INTEGER: 15
SUN-LDOM-MIB::ldomPerfCounters.1 = STRING: htstrand
SUN-LDOM-MIB::ldomPerfCounters.2 = STRING: global,htstrand
SUN-LDOM-MIB::ldomNumCMI.1 = INTEGER: 0
SUN-LDOM-MIB::ldomNumCMI.2 = INTEGER: 0
```

2. *domain-name* ドメインを停止します。

`snmpset` コマンドを使用して、`ldomAdminState` プロパティに値 2 を設定することで、ドメインを停止します。*n* には停止するドメインを指定します。

```
# snmpset -v version -c community-string hostname \
SUN-LDOM-MIB::ldomTable.1.ldomAdminState.n = 2
```

3. 次のいずれかのコマンドを使用して、*domain-name* ドメインがバインドされていることを確認します。

```
■ # ldm list domain-name  
■ # snmpget -v version -c community-string hostname SUN-LDOM-MIB::ldomOperState.n
```

例 21-6 ゲストドメインの停止

この例では、ldomAdminState プロパティを 2 に設定してゲストドメインを停止してから、ldm list LdomMibTest_1 コマンドを使用して LdomMibTest_1 ドメインがバインドされていることを確認しています。

```
# snmpset -v1 -c private localhost SUN-LDOM-MIB::ldomTable.1.ldomAdminState.2 = 2  
# ldm list LdomMibTest_1
```


◆◆◆ 第 22 章

Logical Domains Manager の検出

この章では、サブネット上のシステムで実行されている Logical Domains Manager の検出について説明します。

Logical Domains Manager を実行しているシステムの検出

マルチキャストメッセージを使用すると、サブネット上で Logical Domains Manager を検出できます。ldmd デーモンは、ネットワーク上で特定のマルチキャストパケットを待機できます。そのマルチキャストメッセージが特定のタイプの場合、ldmd は呼び出し元に対して応答します。これにより、Oracle VM Server for SPARC を実行しているシステム上で ldmd を検出できます。

マルチキャスト通信

この検出メカニズムは、ldmd デーモンによって使用されるものと同じマルチキャストネットワークを使用して、MAC アドレスを自動的に割り当てるときに衝突を検出します。マルチキャストソケットを構成するには、次の情報を指定する必要があります。

```
#define MAC_MULTI_PORT 64535
#define MAC_MULTI_GROUP "239.129.9.27"
```

デフォルトでは、マシンが接続されているサブネット上ではマルチキャストパケットのみを送信できます。この動作を、ldmd デーモンに ldmd/hops SMF プロパティを設定することによって変更できます。

メッセージ形式

検出メッセージは、他のメッセージと混同しないように明白にマークされている必要があります。次のマルチキャストメッセージ形式により、検出待機プロセスで検出メッセージを識別できます。

```
#include <netdb.h> /* Used for MAXHOSTNAMELEN definition */
#define MAC_MULTI_MAGIC_NO 92792004
#define MAC_MULTI_VERSION 1

enum {
    SEND_MSG = 0,
    RESPONSE_MSG,
    LDMD_DISC_SEND,
    LDMD_DISC_RESP,
};

typedef struct {
    uint32_t version_no;
    uint32_t magic_no;
    uint32_t msg_type;
    uint32_t resv;
    union {
        mac_lookup_t Mac_lookup;
        ldmd_discovery_t Ldmd_discovery;
    } payload;
#define lookup payload.Mac_lookup
#define discovery payload.Ldmd_discovery
} multicast_msg_t;

#define LDMD_VERSION_LEN 32

typedef struct {
    uint64_t mac_addr;
    char source_ip[INET_ADDRSTRLEN];
} mac_lookup_t;

typedef struct {
    char ldmd_version[LDMD_VERSION_LEN];
    char hostname[MAXHOSTNAMELEN];
    struct in_addr ip_address;
    int port_no;
} ldmd_discovery_t;
```

▼ サブネット上で実行している Logical Domains Manager を検出する方法

1. マルチキャストソケットを開きます。

[521 ページの「マルチキャスト通信」](#)に示すポートおよびグループの情報を使用していることを確認してください。

2. ソケット経由で `multicast_msg_t` メッセージを送信します。

メッセージには次の内容を含めるようにしてください。

- version_no の有効な値 (MAC_MULTI_VERSION によって定義されている 1)
- magic_no の有効な値 (MAC_MULTI_MAGIC_NO によって定義されている 92792004)
- LDMD_DISC_SEND の msg_type

3. マルチキャストソケットで Logical Domains Manager からの応答を待機します。

応答は、次の情報が含まれる multicast_msg_t メッセージである必要があります。

- version_no の有効な値
- magic_no の有効な値
- LDMD_DISC_RESP に設定された msg_type
- 次の情報が含まれる、ldmd_discovery_t 構造で構成されたペイロード
 - ldmd_version – システム上で実行されている Logical Domains Manager のバージョン
 - hostname – システムのホスト名
 - ip_address – システムの IP アドレス
 - port_no – Logical Domains Manager によって通信に使用されているポート番号で、XMPP ポート 6482 にする

Logical Domains Manager からの応答を待機する場合、自動割り当て MAC 衝突検出パケットが破棄されていることを確認してください。

Logical Domains Manager での XML インタフェースの使用

この章では、外部ユーザプログラムが Oracle VM Server for SPARC ソフトウェアと直接対話できる eXtensible Markup Language (XML) の通信メカニズムについて説明します。ここで取り上げる基本事項は、次のとおりです。

- [525 ページの「XML トランスポート」](#)
- [526 ページの「XML プロトコル」](#)
- [532 ページの「イベントメッセージ」](#)
- [538 ページの「Logical Domains Manager のアクション」](#)
- [540 ページの「Logical Domains Manager のリソースとプロパティ」](#)
- [559 ページの「XML スキーマ」](#)

XML トランスポート

外部プログラムは、eXtensible Messaging and Presence Protocol (XMPP – RFC 3920) を使用して、Logical Domains Manager と通信できます。XMPP は、ローカル接続とリモート接続の両方でサポートされており、デフォルトで有効です。リモート接続を無効にするには、ldmd/xmpp_enabled SMF プロパティを false に設定し、Logical Domains Manager を再起動します。

```
# svccfg -s ldom/ldmd setprop ldmd/xmpp_enabled=false
# svcadm refresh ldmd
# svcadm restart ldmd
```

注記 - XMPP サーバーを無効にすると、ドメインの移行およびメモリーの動的再構成も防止されます。

XMPP サーバー

Logical Domains Manager は、数多くの利用可能な XMPP クライアントアプリケーションおよびライブラリと通信できる XMPP サーバーを実装しています。Logical Domains Manager は次のセキュリティーメカニズムを使用しています。

- クライアントと LDoms Manager 自身の間の通信チャネルをセキュリティー保護するための Transport Layer Security (TLS)。
- 認証用の Simple Authentication and Security Layer (SASL)。唯一サポートされている SASL メカニズムは **PLAIN** です。モニタリング操作や管理操作を可能にするには、サーバーが承認できるようにユーザー名およびパスワードをサーバーに送信する必要があります。

ローカル接続

Logical Domains Manager は、ユーザークライアントが LDoms Manager 自身と同じドメインで動作しているかどうかを検出し、同じドメインである場合はこのクライアントとの間で最小限の XMPP ハンドシェークを行います。具体的には、TLS を介したセキュアチャネルの設定後の SASL 認証手順がスキップされます。認証および承認は、クライアントインタフェースを実装しているプロセスの資格に基づいて行われます。

クライアントは、フル XMPP クライアントを実装することも、単に libxml2 Simple API for XML (SAX) パーサーなどのストリーミング XML パーサーを実行することも選択できます。いずれの場合も、クライアントは XMPP ハンドシェークを TLS ネゴシエーションまで処理する必要があります。必要な手順については、XMPP の仕様を参照してください。

XML プロトコル

通信の初期化が完了すると、次に Oracle VM Server for SPARC 定義の XML メッセージが送信されます。XML メッセージの一般的な 2 つのタイプは次のとおりです。

- **<LDM_interface>** タグを使用するリクエストメッセージと応答メッセージ。このタイプの XML メッセージは、コマンドの伝達と、Logical Domains Manager からの結果の取得に使用されます。これはコマンド行インタフェース (CLI) を使用したコマンドの実行に類似しています。このタグは、イベントの登録および登録解除にも使用されます。

- `<LDM_event>` タグを使用するイベントメッセージ。このタイプの XML メッセージは、Logical Domains Manager によって送信されたイベントを非同期に報告するために使用されます。

要求メッセージと応答メッセージ

Oracle VM Server for SPARC の XML インタフェースには、次の異なる 2 つの形式があります。

- Logical Domains Manager にコマンドを送信するための形式。
- 受信メッセージのステータスおよびこのメッセージ内で要求されているアクションに基づいて Logical Domains Manager が応答するための形式。

2 つの形式の XML 構造の多くは共通していますが、両者の違いをよく理解するために、ここでは別々に取り扱います。

要求メッセージ

Logical Domains Manager への受信 XML 要求には、もっとも基本的なレベルで、1 つのオブジェクトで動作する 1 つのコマンドの記述が含まれています。要求が複雑になると、1 つのコマンドで複数のコマンドと複数のオブジェクトを処理できます。次の例に、基本的な XML コマンドの構造を示します。

例 23-1 1 つのオブジェクトで動作する 1 つのコマンドの形式

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <options>Place options for certain commands here</options>
    <arguments>Place arguments for certain commands here</arguments>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content>
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
```

```
        </Section>

        </Content>
    </Envelope>
</data>

</cmd>

</LDM_interface>
```

<LDM_interface> タグ

Logical Domains Manager に送信するすべてのコマンドは、<LDM_interface> タグで始まる必要があります。Logical Domains Manager に送信するドキュメントでは、ドキュメント内に含まれる <LDM_interface> タグは 1 つのみである必要があります。<LDM_interface> タグには、[例23-1「1 つのオブジェクトで動作する 1 つのコマンドの形式」](#)に示すようなバージョン属性が含まれている必要があります。

<cmd> タグ

ドキュメントでは、<LDM_interface> タグ内に 1 つ以上の <cmd> タグが含まれている必要があります。各 <cmd> セクションには、<action> タグを 1 つのみ含める必要があります。この <action> タグは、実行するコマンドを記述するために使用します。各 <cmd> タグに 1 つ以上の <data> タグを含めて、コマンドの処理対象のオブジェクトを記述する必要があります。

また、<cmd> タグには <options> タグも含めることができます。このタグは、一部のコマンドに関連付けられたオプションおよびフラグを指定するために使用されます。次のコマンドにはオプションが使用されます。

- ldm remove-domain コマンドには -a オプションを使用できます。
- ldm bind-domain コマンドには、-f オプションを使用できます。
- ldm add-vdsdev コマンドには、-f オプションを使用できます。
- ldm cancel-operation コマンドには、migration または reconf オプションを使用できます。
- ldm add-spconfig コマンドには、-r *autosave-name* オプションを使用できます。
- ldm remove-spconfig コマンドには、-r オプションを使用できます。
- ldm list-spconfig コマンドには、-r [*autosave-name*] オプションを使用できます。
- ldm stop-domain コマンドでは、次のタグを使用してコマンド引数を設定できます。

- <force> は -f オプションを表します。
- <halt> は -h オプションを表します。
- <message> は -m オプションを表します。
- <quick> は -q オプションを表します。
- <reboot> は -r オプションを表します。
- <timeout> は -t オプションを表します。

このタグにはコンテンツ値を指定できません。ただし、-t および -m オプションには、null 以外の値 (<timeout>10</timeout> や <message>Shutting down now</message> など) を指定する必要があります。

次の XML の例は、リポートメッセージを含むリポートリクエストを `ldm stop-domain` コマンドに渡す方法を示しています。

```
<action>stop-domain</action>
<arguments>
  <reboot/>
  <message>my reboot message</message>
</arguments>
```

<data> タグ

各 <data> セクションには、指定したコマンドに関連するオブジェクトの記述を含めます。<data> セクションの形式は、Open Virtualization Format (OVF) ドラフト仕様の XML スキーマ部分に基づいています。このスキーマは、<References> タグ (Oracle VM Server for SPARC では未使用)、<Content> セクション、および <Section> セクションを含む <Envelope> セクションを定義します。

Oracle VM Server for SPARC の場合、<Content> セクションは、特定のドメインを指定および記述するために使用されます。<Content> ノードの `id=` 属性に指定するドメイン名で、ドメインが識別されます。<Content> セクション内には、特定のコマンドの必要に応じて、ドメインのリソースを記述するための <Section> セクションが 1 つ以上あります。

ドメイン名を指定するだけの場合は、<Section> タグを使用する必要はありません。逆に、コマンドでドメイン識別子が不要な場合は、そのコマンドで必要となるリソースを記述した <Section> セクションを、<Content> セクションの外側で、<Envelope> セクションの内側に配置する必要があります。

オブジェクト情報が推測可能な場合は、<data> セクションに <Envelope> タグを含める必要はありません。この状況は主に、ある処理に該当するすべてのオブジェクトのモニタリング要求、イベントの登録および登録解除の要求に当てはまります。

2 つの追加の OVF タイプによって、OVF 仕様のスキーマを使用して、すべてのタイプのオブジェクトを適切に定義できます。

■ <gprop:GenericProperty> タグ

■ <Binding> タグ

<gprop:GenericProperty> タグは、OVF 仕様に定義がないオブジェクトのプロパティを扱います。プロパティ名はノードの key= 属性に定義され、プロパティの値はノードの内容になります。<binding> タグは、ほかのリソースにバインドされたリソースを定義するために、ldm list-bindings コマンド出力で使用されます。

応答メッセージ

送信 XML 応答は、含まれているコマンドおよびオブジェクトに関して受信要求と厳密に一致した構造を持ちますが、そのほかに、指定されている各オブジェクトおよび各コマンド用の <Response> セクションと、要求に対する全体の <Response> セクションが追加されています。<Response> セクションはステータスおよびメッセージ情報を示します。次の例に、基本的な XML 要求への応答の構造を示します。

例 23-2 1 つのオブジェクトで動作する 1 つのコマンドに対する応答の形式

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content>
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>
                LDom Resource Type
              </rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">
                Property Value
              </gprop:GenericProperty>
            </Item>
```

```

        </Section>
        <!-- Note: More <Section>
    </Content>
</Envelope>
<response>
    <status>success or failure</status>
    <resp_msg>Reason for failure</resp_msg>
</response>
</data>

<response>
    <status>success or failure</status>
    <resp_msg>Reason for failure</resp_msg>
</response>
</cmd>

<response>
    <status>success or failure</status>
    <resp_msg>Reason for failure</resp_msg>
</response>
</LDM_interface>

```

全体の応答

この `<response>` セクションは、`<LDM_interface>` セクションの直下の子であり、要求全体の成功または失敗を示します。受信 XML ドキュメントが不正な形式でないかぎり、`<response>` セクションには、`<status>` タグだけが含まれます。この応答ステータスが成功を示している場合、すべてのオブジェクトに対するすべてのコマンドが成功しています。この応答ステータスが失敗を示し、`<resp_msg>` タグがない場合は、元の要求内のコマンドのいずれかが失敗しています。`<resp_msg>` タグは、XML ドキュメント自体の問題を記述する場合にのみ使用されます。

コマンドの応答

`<cmd>` セクションの下にある `<response>` セクションは、特定のコマンドの成功または失敗についてユーザーに通知します。`<status>` タグは、このコマンドが成功したか失敗したかを示します。全体の応答の場合と同様に、コマンドが失敗した場合で、要求の `<cmd>` セクションの内容の形式が不正なときは、`<response>` セクションには `<resp_msg>` タグのみが含まれます。それ以外の場合の失敗ステータスは、コマンドが実行されたオブジェクトのいずれかが原因で失敗したことを示しています。

オブジェクトの応答

最後に、<cmd> セクション内の各 <data> セクションにも、<response> セクションがあります。このセクションでは、この特定のオブジェクトで実行されたコマンドが成功したか失敗したかを示します。応答のステータスが SUCCESS の場合、<response> セクション内に <resp_msg> タグはありません。ステータスが FAILURE の場合、そのオブジェクトでのコマンドの実行時に発生したエラーに応じて、<response> フィールドには 1 つ以上の <resp_msg> タグがあります。オブジェクトエラーは、コマンドの実行時に検出された問題、または不正な形式または不明なオブジェクトが原因で発生する可能性があります。

<response> セクションのほかに、<data> セクションにその他の情報が含まれていることがあります。この情報は、受信 <data> フィールドと同じ形式で、失敗の原因となったオブジェクトを記述しています。[529 ページの「<data> タグ」](#)を参照してください。この追加情報は、次の場合に特に有用です。

- コマンドの実行が、特定の <data> セクションに対して失敗したが、別の <data> セクションに対しては成功した場合
- 空の <data> セクションがコマンドに渡されて、一部のドメインでは実行に失敗したが、ほかのドメインでは成功した場合

イベントメッセージ

ポーリングの代わりに、特定の状態変化が発生した場合にイベント通知を受信するように登録できます。個々に、または一括して登録できるイベントのタイプは 3 つあります。詳細は、[534 ページの「イベントタイプ」](#)を参照してください。

登録および登録解除

イベントを登録するには、<LDM_interface> メッセージを使用します。[528 ページの「<LDM_interface> タグ」](#)を参照してください。<action> タグには登録または登録解除するイベントのタイプを記述し、<data> セクションは空白のままにしておきます。

例 23-3 イベントの登録要求メッセージの例

```
<LDM_interface version="1.3">
```

```

<cmd>
  <action>reg-domain-events</action>
  <data version="3.0"/>
</cmd>
</LDM_interface>

```

Logical Domains Manager は、登録または登録解除が成功したかどうかを示す `<LDM_interface>` 応答メッセージで応答します。

例 23-4 イベントの登録応答メッセージの例

```

<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
    <response>
      <status>SUCCESS</status>
    </response>
  </data>
  <response>
    <status>SUCCESS</status>
  </response>
</cmd>
<response>
  <status>SUCCESS</status>
</response>
</LDM_interface>

```

各タイプのイベントの処理文字列は、イベントサブセクションにリストされます。

<LDM_event> メッセージ

イベントメッセージの形式は受信 `<LDM_interface>` メッセージと同じですが、このメッセージの開始タグは `<LDM_event>` になる点が異なります。メッセージの `<action>` タグは、イベントをトリガーするために実行されたアクションです。メッセージの `<data>` セクションにはイベントに関連付けられたオブジェクトが記述されます。詳細は、発生したイベントのタイプによって異なります。

例 23-5 <LDM_event> 通知の例

```

<LDM_event version='1.1'>
  <cmd>
    <action>Event command here</action>
    <data version='3.0'>

```

```
<Envelope
  <References/>
  <Content xsi:type='ovf:VirtualSystem_Type' ovf:id='ldg1' />
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>
      <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
      <gprop:GenericProperty
        key="Property name">Property Value</gprop:GenericProperty>
      </Item>
    </Section>
  </Envelope>
</data>
</cmd>
</LDM_event>
```

イベントタイプ

次のイベントタイプを登録できます。

- ドメインイベント
- ハードウェアイベント
- 進捗イベント
- リソースイベント

すべてのイベントが `ldm` サブコマンドに対応しています。

ドメインイベント

ドメインイベントは、ドメインに直接実行できるアクションを記述します。次のドメインイベントを、`<LDM_event>` メッセージの `<action>` タグに指定することができます。

- `add-domain`
- `bind-domain`
- `domain-reset`
- `migrate-domain`
- `panic-domain`
- `remove-domain`
- `start-domain`
- `stop-domain`

■ unbind-domain

これらのイベントでは、常に、OVF <data> セクションに、イベントが発生したドメインを記述する <Content> タグのみが含まれます。ドメインイベントを登録するには、<action> タグを reg-domain-events に設定した <LDM_interface> メッセージを送信します。これらのイベントを登録解除するには、<action> タグを unreg-domain-events に設定した <LDM_interface> メッセージを送信します。

ハードウェアイベント

ハードウェアイベントは、物理的なシステムハードウェアの変更に関係しています。Oracle VM Server for SPARC ソフトウェアの場合、ハードウェア変更は、サービスプロセッサ (SP) 構成の追加、削除、または設定を行う場合の SP への変更だけです。現在、このタイプのイベントは次の 3 つだけです。

■ add-spconfig

■ set-spconfig

■ remove-spconfig

ハードウェアイベントでは、常に、OVF <data> セクションにイベントが発生している SP 構成を記述する <Section> タグのみが含まれます。これらのイベントを登録するには、<action> タグを reg-hardware-events に設定した <LDM_interface> メッセージを送信します。これらのイベントを登録解除するには、<action> タグを unreg-hardware-events に設定した <LDM_interface> メッセージを送信します。

進捗イベント

進捗イベントは、ドメインの移行など、長時間にわたって実行されるコマンドに対して発行されます。このイベントは、コマンド実行期間中のそれまでの進捗量を報告します。この時点では、migration-process イベントのみが報告されます。

進捗イベントでは、常に、OVF <data> セクションにイベントの影響を受ける SP 構成が記述された <Section> タグのみが含まれます。これらのイベントを登録するには、<action> タグを reg-hardware-events に設定した <LDM_interface> メッセージを送信します。これらのイベントを登録解除するには、<action> タグを unreg-hardware-events に設定した <LDM_interface> メッセージを送信します。

進捗イベントの <data> セクションは、影響を受けるドメインを記述する <content> セクションによって構成されています。この <content> セクションでは、ldom_info <Section> タグを使用して進捗を更新します。次の汎用プロパティが ldom_info セクションに表示されます。

- --progress – コマンドの進捗の割合
- --status – コマンドのステータス。ongoing、failed、または done のいずれか
- --source – 進捗を報告しているマシン

リソースイベント

任意のドメインでリソースを追加、削除、または変更すると、リソースイベントが発生します。これらの一部のイベントの<data> セクションには、OVF <data> セクションにサービス名が示されている <Section> タグがある、<Content> タグが含まれています。

次のイベントを、<LDM_event> メッセージの <action> タグに指定することができます。

- add-vdiskserverdevice
- remove-vdiskserverdevice
- set-vdiskserverdevice
- remove-vdiskserver
- set-vconscon
- remove-vconscon
- set-vswitch
- remove-vswitch
- remove-vdpcs

次のリソースイベントでは、常に、OVF <data> セクションに、イベントの発生したドメインが記述された <Content> タグのみが含まれます。

- add-vcpu
- add-crypto
- add-memory
- add-io
- add-variable
- add-vconscon
- add-vdisk

- add-vdiskserver
- add-vnet
- add-vswitch
- add-vdpcs
- add-udpcc
- set-vcpu
- set-crypto
- set-memory
- set-variable
- set-vnet
- set-vconsole
- set-vdisk
- remove-vcpu
- remove-crypto
- remove-memory
- remove-io
- remove-variable
- remove-vdisk
- remove-vnet
- remove-udpcc

リソースイベントを登録するには、<action> タグを `reg-resource-events` に設定した <LDM_interface> メッセージを送信します。これらのイベントの登録を解除するには、<action> タグを `unreg-resource-events` に設定した <LDM_interface> メッセージが必要です。

すべてのイベント

各イベントを個別に登録しないで、3 つのタイプすべてのイベントを待機するように登録することもできます。3 タイプすべてのイベントを同時に登録するには、<action> タグを `reg-all-events` に設定した <LDM_interface> メッセージを送信します。これらのイベントの登録を解除するには、<action> タグを `unreg-all-events` に設定した <LDM_interface> メッセージが必要です。

Logical Domains Manager のアクション

<action> タグに指定するコマンドは、*-*-events コマンドを除いて、ldm コマンド行インタフェースのコマンドに対応しています。ldm サブコマンドの詳細については、[ldm\(1M\)](#) マニュアルページを参照してください。

注記 - XML インタフェースは、Logical Domains Manager CLI でサポートされている動詞またはコマンドの別名はサポートしていません。

<action> タグでサポートされている文字列は、次のとおりです。

- add-domain
- add-io
- add-mau
- add-memory
- add-spconfig
- add-variable
- add-vconscon
- add-vcpu
- add-vdisk
- add-vdiskserver
- add-vdiskserverdevice
- add-vdpc
- add-vdpcs
- add-vnet
- add-vswitch
- bind-domain
- cancel-operation
- list-bindings
- list-constraints
- list-devices
- list-domain

- list-services
- list-spconfig
- list-variable
- migrate-domain
- reg-all-events
- reg-domain-events
- reg-hardware-events
- reg-resource-events
- remove-domain
- remove-io
- remove-mau
- remove-memory
- remove-reconf
- remove-spconfig
- remove-variable
- remove-vconscon
- remove-vcpu
- remove-vdisk
- remove-vdiskserver
- remove-vdiskserverdevice
- remove-udpcc
- remove-udpccs
- remove-vnet
- remove-vswitch
- set-domain
- set-mau
- set-memory
- set-spconfig
- set-variable
- set-vconscon
- set-vconsole

- set-vcpu
- set-vnet
- set-vswitch
- start-domain
- stop-domain
- unbind-domain
- unreg-all-events
- unreg-domain-events
- unreg-hardware-events
- unreg-resource-events

Logical Domains Manager のリソースとプロパティ

このセクションでは、Logical Domains Manager リソースの例と、それらの各リソースに定義できるプロパティについて説明します。XML の例では、リソースおよびプロパティは太字で示されています。これらの例は、バインド出力ではなくリソースを示しています。制約出力は、ドメイン移行の出力を除いて、Logical Domains Manager のアクションの入力を作成する場合に使用できます。[558 ページの「ドメインの移行」](#)を参照してください。各リソースは、<Section> の OVF セクションで定義され、<rasd:OtherResourceType> タグによって指定されます。

ドメインの情報 (ldom_info) リソース

次の例に、ldom_info リソースのオプションのプロパティを示します。

例 23-6 ldom_info の XML 出力の例

次の例に、uuid、hostid、Address など、複数の ldom_info プロパティに指定された値を示します。

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="primary">
    <Section xsi:type="ovf:ResourceAllocationSection_type">
```

```

<Item>
  <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
  <uuid>c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6</uuid>
  <rasd:Address>00:03:ba:d8:ba:f6</rasd:Address>
  <gprop:GenericProperty key="hostid">83d8baf6</gprop:GenericProperty>
  <gprop:GenericProperty key="master">plum</gprop:GenericProperty>
  <gprop:GenericProperty key="failure-policy">reset</gprop:GenericProperty>
  <gprop:GenericProperty key="extended-mapin-space">on</gprop:GenericProperty>
  <gprop:GenericProperty key="progress">45%</gprop:GenericProperty>
  <gprop:GenericProperty key="status">ongoing</gprop:GenericProperty>
  <gprop:GenericProperty key="source">system1</gprop:GenericProperty>
  <gprop:GenericProperty key="rc-add-policy"></gprop:GenericProperty>
  <gprop:GenericProperty key="perf-counters">global</gprop:GenericProperty>
</Item>
</Section>
</Content>
</Envelope>

```

ldom_info リソースは、<Content> セクション内に必ず含まれます。ldom_info リソース内の次のプロパティは、オプションです。

- <uuid> タグ。ドメインの UUID を指定します。
- <rasd:Address> タグ。ドメインに割り当てる MAC アドレスを指定します。
- <gprop:GenericProperty key="extended-mapin-space"> タグ。これは、ドメインの拡張されたマップイン領域を有効 (on) にするか、または無効 (off) にするかを指定します。デフォルト値は off です。
- <gprop:GenericProperty key="failure-policy"> タグ。マスタートメインに障害が発生した場合のスレーブドメインの動作を指定します。デフォルト値は ignore です。次に、有効なプロパティ値を示します。
 - ignore は、マスタートメインの障害を無視します。スレーブドメインは影響を受けません。
 - panic は、マスタートメインに障害が発生した場合、すべてのスレーブドメインにパニックを発生させます。
 - reset は、マスタートメインに障害が発生した場合、すべてのスレーブドメインをリセットします。
 - stop は、マスタートメインに障害が発生した場合、すべてのスレーブドメインを停止します。
- <gprop:GenericProperty key="hostid"> タグ。ドメインに割り当てるホスト ID を指定します。
- <gprop:GenericProperty key="master"> タグ。最大 4 つのマスタートメイン名をコンマで区切って指定します。
- <gprop:GenericProperty key="progress"> タグ。コマンドの進捗の割合を指定します。

- `<gprop:GenericProperty key="source">` タグ。コマンドの進捗を報告するマシンを指定します。
- `<gprop:GenericProperty key="status">` タグ。コマンドのステータス (done, failed、または ongoing) を指定します。
- `<gprop:GenericProperty key="rc-add-policy">` タグは、特定のドメインに追加される可能性のあるルートコンプレックスで、直接 I/O および SR-IOV I/O 仮想化操作を有効または無効にするかどうかを指定します。有効な値は、iov および値なし (rc-add-policy=) です。
- アクセスするパフォーマンスレジスタセット (global, htstrand, strand) を指定する、`<gprop:GenericProperty key="perf-counters">` タグ。
プラットフォームにパフォーマンスアクセス機能がない場合、perf-counters プロパティ値は無視されます。

CPU (cpu) リソース

cpu リソースの割り当て単位プロパティの `<rasd:AllocationUnits>` では、コアの数ではなく仮想 CPU の数を常に指定します。

cpu リソースは、`<Content>` セクション内に必ず含まれます。

例 23-7 `ldm list-bindings` コマンドによる cpu XML セクションの出力

次の例では、`ldm list-bindings` コマンドの使用による `<cpu>` セクションの XML 出力を示します。

```
<?xml version="1.0"?>
<LDM_interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding"
version="1.3"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd">
  <cmd>
    <action>list-bindings</action>
    <data version="3.0">
      <Envelope>
        <References/>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

```

<Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
  <Section xsi:type="ovf:ResourceAllocationSection_Type">
    <Item>
      <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
      <uuid>1e04cddb-472a-e8b9-ba4c-d3eee86e7725</uuid>
      <rasd:Address>00:21:28:f5:11:6a</rasd:Address>
      <gprop:GenericProperty key="hostid">0x8486632a</gprop:GenericProperty>
      <failure-policy>fff</failure-policy>
      <wcore>0</wcore>
      <extended-mapin-space>0</extended-mapin-space>
      <cpu-arch>native</cpu-arch>
      <rc-add-policy/>
      <gprop:GenericProperty key="state">active</gprop:GenericProperty>
    </Item>
  </Section>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
      <rasd:AllocationUnits>8</rasd:AllocationUnits>
      <bind:Binding>
        <Item>
          <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
          <gprop:GenericProperty key="vid">0</gprop:GenericProperty>
          <gprop:GenericProperty key="pid">0</gprop:GenericProperty>
          <gprop:GenericProperty key="cid">0</gprop:GenericProperty>
          <gprop:GenericProperty key="strand_percent">100</gprop:GenericProperty>
          <gprop:GenericProperty key="util_percent">1.1%</gprop:GenericProperty>
          <gprop:GenericProperty key="normalized_utilization">0.1%</gprop:
GenericProperty>
        </Item>
      </Section>
    </Content>
  </Envelope>
</data>
</cmd>
</LDM_interface>

```

例 23-8 ldm list-domain コマンドによる cpu XML セクションの出力

次の例では、ldm list-domain コマンドの使用による <cpu> セクションの XML 出力を示します。

```

<?xml version="1.0"?>
<LDM_interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="http://schemas.dmtf.org/ovf/envelope"
xmlns:rasd="http://schemas.dmtf.org/ovf/resourceallocation"
xmlns:vssd="http://schemas.dmtf.org/ovf/virtualsystem"
xmlns:gprop="http://schemas.dmtf.org/ovf/genericproperty"
xmlns:bind="http://schemas.dmtf.org/ovf/binding"
version="1.3"
xsi:noNamespaceSchemaLocation="http://schemas.dmtf.org/ovf/combined-v3.xsd">

```

```

<cmd>
  <action>list-domain</action>
  <data version="3.0">
    <Envelope>
      <References/>
      <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
        <Section xsi:type="ovf:ResourceAllocationSection_Type">
          <Item>
            <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
            <gprop:GenericProperty key="state">active</gprop:GenericProperty>
            <gprop:GenericProperty key="flags">-n-cv-</gprop:GenericProperty>
            <gprop:GenericProperty key="utilization">0.7%</gprop:GenericProperty>
            <gprop:GenericProperty key="uptime">3h</gprop:GenericProperty>
            <gprop:GenericProperty key="normalized_utilization">0.1%</gprop:GenericProperty>
          </Item>
        </Section>
      </Content>
    </Envelope>
  </data>
</cmd>
</LDM_interface>

```

MAU (mau) リソース

mau リソースは、<Content> セクション内に必ず含まれます。プロパティは <rasd: AllocationUnits> タグのみで、MAU またはその他の暗号化装置の数を指定します。

注記 - mau リソースは、サポートされているサーバーでサポートする暗号化装置です。現在、モジュラー演算ユニット (Modular Arithmetic Unit, MAU) と Control Word Queue (CWQ) の 2 つの暗号化装置がサポートされています。

例 23-9 mau の XML の例

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>mau</rasd:OtherResourceType>
        <rasd:AllocationUnits>1</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>

```


メモリー (memory) リソース

メモリーリソースは、<Content> セクション内に必ず含まれます。プロパティは <rasd:AllocationUnits> タグのみで、メモリーの量を指定します。

例 23-10 memory の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>memory</rasd:OtherResourceType>
        <rasd:AllocationUnits>4G</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

仮想ディスクサーバー (vds) リソース

仮想ディスクサーバー (vds) リソースは、ドメイン記述の一部として <Content> セクションに含まれることも、単独で <Envelope> セクションに記述されることもあります。プロパティは <gprop:GenericProperty> タグのみです。このタグには、"service_name" というキーがあり、記述される vds リソースの名前が含まれています。

例 23-11 vds の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vds</rasd:OtherResourceType>
        <gprop:GenericProperty
          key="service_name">vdstmp</gprop:GenericProperty>
        </Item>
      </Section>
    </Content>
  </Envelope>
```

仮想ディスクサーバーボリューム (vds_volume) リソース

vds_volume リソースは、ドメイン記述の一部として <Content> セクションに含まれることも、単独で <Envelope> セクションに記述されることもあります。次のキーを持つ <gprop:GenericProperty> タグが必要です。

- vol_name – ボリュームの名前
- service_name – このボリュームをバインドする仮想ディスクサーバーの名前
- block_dev – このボリュームに関連付けるファイルまたはデバイスの名前

オプションで、vds_volume リソースに次のプロパティも設定できます。

- vol_opts – {ro,slice,excl} のように、これらの項目の 1 つ以上がコンマで区切られて、1 つの文字列となっているもの
- mpgroup – マルチパス (フェイルオーバー) グループの名前

例 23-12 vds_volume の XML の例

```
<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vds_volume</rasd:OtherResourceType>
      <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
      <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
      <gprop:GenericProperty key="block_dev">
        opt/SUNWldm/domain_disks/testdisk1</gprop:GenericProperty>
      <gprop:GenericProperty key="vol_opts">ro</gprop:GenericProperty>
      <gprop:GenericProperty key="mpgroup">mpgroup-name</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

ディスク (disk) リソース

disk リソースは、<Content> セクション内に必ず含まれます。次のキーを持つ <gprop:GenericProperty> タグが必要です。

- vdisk_name – 仮想ディスクの名前
- service_name – この仮想ディスクをバインドする仮想ディスクサーバーの名前
- vol_name – この仮想ディスクに関連付ける仮想ディスクサービスデバイス

オプションで、disk リソースに timeout プロパティも含めることができます。このプロパティは、仮想ディスククライアント (vdc) と仮想ディスクサーバー (vds) の間に接続を確立するためのタイムアウト値です (秒単位)。複数の仮想ディスク (vdisk) パスがある場合、vdc は、別の vds への接続を試みることができます。タイムアウトにより、すべての vds への接続が、指定した時間内に確立されます。

例 23-13 disk の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>disk</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdisk_name">vdisk0</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">primary-vds0</gprop:GenericProperty>
        <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
        <gprop:GenericProperty key="timeout">60</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

仮想スイッチ (vsw) リソース

vsw リソースは、ドメイン記述の一部として <Content> セクションに含まれることも、単独で <Envelope> セクションに記載されることもあります。それには、service_name キーがある <gprop:GenericProperty> タグが必要です。その名前が仮想スイッチに割り当てられます。オプションで、vsw リソースに次のプロパティも設定できます。

- <rasd:Address> – MAC アドレスを仮想スイッチに割り当てます。
- default-vlan-id – 仮想ネットワークデバイスまたは仮想スイッチをメンバーにする必要があるデフォルトの仮想ローカルエリアネットワーク (VLAN) を、タグ付きモードで指定します。最初の VLAN ID (vid1) は、この default-vlan-id に予約されています。
- dev_path – この仮想スイッチに関連付けるネットワークデバイスのパス
- id – 新しい仮想スイッチデバイスの ID を指定します。デフォルトでは ID 値は自動的に生成されるため、OS で既存のデバイス名に一致させる必要がある場合に、このプロパティを設定します。

- **inter_vnet_link** – inter-vnet 通信用の LDC チャンネルを割り当てるかどうかを指定します。デフォルト値は on です。
- **linkprop** – 仮想デバイスが物理リンクステータスの更新を取得するかどうかを指定します。値が **phys-state** の場合、仮想デバイスは物理リンクステータスの更新を取得します。値が空白の場合、仮想デバイスは物理リンク状態の更新を取得しません (デフォルトの設定)。
- **mode** – Oracle Solaris Cluster のハートビートサポートの場合は **sc**。
- **pvid** – ポート仮想ローカルエリアネットワーク (VLAN) 識別子 (ID)。仮想ネットワークをメンバーにする必要のある VLAN をタグなしモードで指定します。
- **mtu** – 仮想スイッチと、その仮想スイッチにバインドされている仮想ネットワークデバイス、またはその両方に最大転送単位 (Maximum Transmission Unit, MTU) を指定します。有効な値の範囲は 1500 - 16000 です。無効な値を指定すると、ldm コマンドでエラーが発生します。
- **vid** – 仮想ローカルエリアネットワーク (VLAN) 識別子 (ID)。仮想ネットワークおよび仮想スイッチをメンバーにする必要のある VLAN をタグ付きモードで指定します。

例 23-14 vsw の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg2">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vsw</rasd:OtherResourceType>
        <rasd:Address>00:14:4f:fb:ec:00</rasd:Address>
        <gprop:GenericProperty key="service_name">test-vsw1</gprop:GenericProperty>
        <gprop:GenericProperty key="inter_vnet_link">on</gprop:GenericProperty>
        <gprop:GenericProperty key="default-vlan-id">1</gprop:GenericProperty>
        <gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
        <gprop:GenericProperty key="mtu">1500</gprop:GenericProperty>
        <gprop:GenericProperty key="dev_path">switch@0</gprop:GenericProperty>
        <gprop:GenericProperty key="id">0</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

ネットワーク (network) リソース

network リソースは、<Content> セクション内に必ず含まれます。次のキーを持つ <gprop:GenericProperty> タグが必要です。

- linkprop – 仮想デバイスが物理リンクステータスの更新を取得するかどうかを指定します。値が phys-state の場合、仮想デバイスは物理リンクステータスの更新を取得します。値が空白の場合、仮想デバイスは物理リンク状態の更新を取得しません (デフォルトの設定)。
- vnet_name – 仮想ネットワーク (vnet) の名前
- service_name – この仮想ネットワークをバインドする仮想スイッチ (vswitch) の名前

オプションで、network リソースに次のプロパティも設定できます。

- <rasd:Address> – MAC アドレスを仮想スイッチに割り当てます。
- pvid – ポート仮想ローカルエリアネットワーク (VLAN) 識別子 (ID)。仮想ネットワークをメンバーにする必要のある VLAN をタグなしモードで指定します。
- vid – 仮想ローカルエリアネットワーク (VLAN) 識別子 (ID)。仮想ネットワークおよび仮想スイッチをメンバーにする必要のある VLAN をタグ付きモードで指定します。
- mode – 仮想ネットワークに対してハイブリッド I/O を有効にする場合は hybrid。

注記 - SR-IOV が使われるようになって NIU Hybrid I/O 機能は非推奨になりました。Oracle VM Server for SPARC 3.3 は、この機能を含む最後のソフトウェアリリースです。

例 23-15 network の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>network</rasd:OtherResourceType>
        <gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
        <gprop:GenericProperty key="vnet_name">ldg1-vnet0</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">primary-vsw0</gprop:GenericProperty>
        <rasd:Address>00:14:4f:fc:00:01</rasd:Address>
      </Item>
    </Section>
  </Content>
</Envelope>
```

仮想コンソール端末集配信装置 (vcc) リソース

vcc リソースは、ドメイン記述の一部として <Content> セクションに含まれることも、単独で <Envelope> セクションに記述されることもあります。次のキーを持つ <gprop:GenericProperty> タグを使用できます。

- service_name – 仮想コンソール端末集配信装置サービスに割り当てる名前
- min_port – この vcc に関連付ける最小ポート番号
- max_port – この vcc に関連付ける最大ポート番号

例 23-16 vcc の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vcc</rasd:OtherResourceType>
        <gprop:GenericProperty key="service_name">vcc1</gprop:GenericProperty>
        <gprop:GenericProperty key="min_port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="max_port">6100</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

変数 (var) リソース

var リソースは、<Content> セクション内に必ず含まれます。次のキーを持つ <gprop:GenericProperty> タグを使用できます。

- name – 変数の名前
- value – 変数の値

例 23-17 var の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>var</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">test_var</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

```

        <gprop:GenericProperty key="value">test1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

物理 I/O デバイス (physio_device) リソース

physio_device リソースは、<Content> セクション内に必ず含まれます。このリソースは、add-io、set-io、remove-io、create-vf、destroy-vf、および set-domain サブコマンドを使用して変更できます。

例 23-18 physio_device の XML の例

次の例は、仮想機能、物理機能、ルートコンプレックスでアクションを実行する方法を示します。

- 次の XML の例では、ldm add-io コマンドを使用して、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を ldg1 ドメインに追加する方法を示します。

```

<LDM_interface version="1.3">
  <cmd>
    <action>add-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1">
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
              <gprop:GenericProperty key="name">
                /SYS/MB/NET0/IOVNET.PF0.VF0</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

- 次の XML の例に、ldm set-io コマンドを使用して、pci_1 ルートコンプレックスの iov_bus_enable_iov プロパティ値を on に設定する方法を示します。

```
<LDM_interface version="1.3">
  <cmd>
    <action>set-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="name">pci_1</gprop:GenericProperty>
            <gprop:GenericProperty key="iov_bus_enable_iov">
              on</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

- 次の XML の例に、ldm set-io コマンドを使用して、/SYS/MB/NET0/IOVNET.PF1 物理機能の unicast-slots プロパティ値を 6 に設定する方法を示します。

```
<LDM_interface version="1.3">
  <cmd>
    <action>set-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="unicast-slots">6</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```



```

        </Section>
    </Envelope>
</data>
</cmd>
</LDM_interface>

```

- 次の XML の例では、`ldm create-vf` コマンドを使用して、次のプロパティ値で `/SYS/MB/NET0/IOVNET.PF1.VF0` 仮想機能を作成する方法を示します。

- `unicast-slots=6`
- `pvid=3`
- `mtu=1600`

```

<LDM_interface version="1.3">
  <cmd>
    <action>create-vf</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>vf_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="iov_pf_name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="unicast-slots">6</gprop:GenericProperty>
            <gprop:GenericProperty key="pvid">3</gprop:GenericProperty>
            <gprop:GenericProperty key="mtu">1600</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

- 次の部分的な XML の例は、`ldm create-vf` コマンドを使用して、`/SYS/MB/NET0/IOVNET.PF1` 物理機能で `iov_pf_repeat_count_str` 値 (3) によって指定された数の仮想機能を作成する方法を示します。`iov_pf_repeat_count_str` プロパティで複数の仮想機能を作成するときにほかのプロパティ値を指定することはできません。

```

<LDM_interface version="1.3">
  <cmd>
    <action>create-vf</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>vf_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="iov_pf_name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="iov_pf_repeat_count_str">
              3</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

SP 構成 (spconfig) リソース

サービスプロセッサ (SP) 構成 (spconfig) リソースは、必ず単独で <Envelope> セクションに記述されます。次のキーを持つ <gprop:GenericProperty> タグを使用できます。

- spconfig_name – SP に格納されている構成の名前。
- spconfig_status – 特定の SP 構成の現在のステータス。このプロパティは、ldm list-spconfig コマンドの出力で使われます。

例 23-19 spconfig の XML の例

```

<Envelope>
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>
      <rasd:OtherResourceType>spconfig</rasd:OtherResourceType>
      <gprop:GenericProperty
        key="spconfig_name">primary</gprop:GenericProperty>
      <gprop:GenericProperty
        key="spconfig_status">current</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

```

    </Item>
  </Section>
</Envelope>

```

DRM ポリシー構成 (policy) リソース

DRM ポリシー (policy) リソースは、<Envelope> セクションに表示され、次のキーが付いた<gprop:GenericProperty> タグを持つことができます。

- policy_name – DRM ポリシーの名前
- policy_enable – DRM ポリシーが有効または無効のどちらであるかを指定します。
- policy_priority – DRM ポリシーの優先順位
- policy_vcpu_min – 1 つのドメインに対する仮想 CPU リソースの最小数
- policy_vcpu_max – 1 つのドメインに対する仮想 CPU リソースの最大数
- policy_util_lower – ポリシー分析がトリガーされるタイミングでの使用率の下限レベル
- policy_util_upper – ポリシー分析がトリガーされるタイミングでの使用率の上限レベル
- policy_tod_begin – DRM ポリシーの実効開始時間
- policy_tod_end – DRM ポリシーの実効停止時間
- policy_sample_rate – サンプリングレート。秒単位のサイクル時間
- policy_elastic_margin – CPU 使用率の上限および下限範囲のバッファ量
- policy_attack – いずれか 1 つのリソース制御サイクル中に追加されるリソースの最大量
- policy_decay – いずれか 1 つのリソース制御サイクル中に削除されるリソースの最大量

例 23-20 policy の XML の例

```

<Envelope>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>policy</rasd:OtherResourceType>
      <gprop:GenericProperty key="policy_name">test-policy</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_enable">on</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_priority">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_min">12</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_max">13</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_lower">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_upper">9</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_begin">07:08:09</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_end">09:08:07</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_sample_rate">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_elastic_margin">8</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

```

      <gprop:GenericProperty key="policy_attack">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_decay">9</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

仮想データプレーンチャンネルサービス (vdpcs) リソース

このリソースは、Netra DPS 環境でのみ意味を持ちます。vdpcs リソースは、ドメイン記述の一部として <Content> セクションに含まれることも、単独で <Envelope> セクションに記述されることもあります。プロパティは、<gprop:GenericProperty> タグのみです。このタグには、"service_name" というキープロパティ値があり、記述される仮想データプレーンチャンネルサービス (vdpcs) リソースの名前が含まれています。

例 23-21 vdpcs の XML の例

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vdpcs</rasd:OtherResourceType>
        <gprop:GenericProperty key="service_name">dg1-vdpcs</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

仮想データプレーンチャンネルクライアント (vdpsc) リソース

このリソースは、Netra DPS 環境でのみ意味を持ちます。仮想データプレーンチャンネルクライアントリソースは、<Content> セクション内に必ず含まれます。次のキーを持つ <gprop:GenericProperty> タグを使用できます。

- vdpsc_name – 仮想データプレーンチャンネルクライアント (vdpsc) の名前
- service_name – この vdpsc をバインドする仮想データプレーンチャンネルサービス vdpcs の名前

例 23-22 vdpsc の XML の例

```

<Envelope>

```

```

<References/>
<Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vdpc</rasd:OtherResourceType>
      <gprop:GenericProperty key="vdpc_name">vdpc</gprop:GenericProperty>
      <gprop:GenericProperty
        key="service_name">ldg1-vdpcs</gprop:GenericProperty>
    </Item>
  </Section>
</Content>
</Envelope>

```

コンソール (console) リソース

console リソースは、<Content> セクション内に必ず含まれます。次のキーを持つ <gprop:GenericProperty> タグを使用できます。

- port – この仮想コンソール (console) の変更先のポート
- service_name – この console をバインドする仮想コンソール端末集配信装置 (vcc) サービス
- group – この console をバインドするグループの名前
- enable-log – このコンソールの仮想コンソールログギングを有効または無効にします

例 23-23 console の XML の例

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>console</rasd:OtherResourceType>
        <gprop:GenericProperty key="port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">vcc2</gprop:GenericProperty>
        <gprop:GenericProperty key="group">group-name</gprop:GenericProperty>
        <gprop:GenericProperty key="enable-log">on</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

ドメインの移行

次の例は、ldm migrate-domain コマンドの <data> セクションの内容を示しています。

- 1 番めの <Content> ノード (<ldom_info> セクションなし) は、移行元のソースドメインです。
- 2 番めの <Content> ノード (<ldom_info> セクションあり) は、移行先のターゲットドメインです。ソースドメインとターゲットドメインの名前は同じにすることができます。
- ターゲットドメインの <ldom_info> セクションには、移行先のマシンおよびこのマシンへの移行に必要な詳細情報が記述されます。
 - target-host は、移行先のターゲットマシンです。
 - user-name は、ターゲットマシンのログインユーザー名で、SASL 64 ビットでエンコードされている必要があります。
 - password は、ターゲットマシンへのログインに使用するパスワードで、SASL 64 ビットでエンコードされている必要があります。

注記 - Logical Domains Manager では、sas_l_decode64() を使用してターゲットのユーザー名およびパスワードを復号化し、sas_l_encode64() を使用してこれらの値をエンコードします。SASL 64 符号化は、base64 符号化に相当します。

例 23-24 migrate-domain の <data> セクションの例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
    <Section xsi:type="ovf:ResourceAllocationSection_Type">
      <Item>
        <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
        <gprop:GenericProperty key="target">target-host</gprop:GenericProperty>
        <gprop:GenericProperty key="username">user-name</gprop:GenericProperty>
        <gprop:GenericProperty key="password">password</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

XML スキーマ

Logical Domains Manager で使用される XML スキーマは、`/opt/SUNWldm/bin/schemas` ディレクトリにあります。ファイル名は、次のとおりです。

- `cim-common.xsd` – `cim-common.xsd` スキーマ
- `cim-rasd.xsd` – `cim-rasd.xsd` スキーマ
- `cim-vssd.xsd` – `cim-vssd.xsd` スキーマ
- `cli-list-constraint-v3.xsd` – `cli-list-constraint-v3.xsd` スキーマ
- `combined-v3.xsd` – `LDM_interface XML` スキーマ
- `event-v3.xsd` – `LDM_Event XML` スキーマ
- `ldmd-binding.xsd` – `Binding_Type XML` スキーマ
- `ldmd-property.xsd` – `GenericProperty XML` スキーマ
- `ovf-core.xsd` – `ovf-core.xsd` スキーマ
- `ovf-envelope.xsd` – `ovf-envelope.xsd` スキーマ
- `ovf-section.xsd` – `ovf-section.xsd` スキーマ
- `ovf-strings.xsd` – `ovf-strings.xsd` スキーマ
- `ovfenv-core.xsd` – `ovfenv-core.xsd` スキーマ
- `ovfenv-section.xsd` – `ovfenv-section.xsd` スキーマ

用語集

この用語集では、Oracle VM Server for SPARC のドキュメントで使用されている用語、略語、および頭字語を定義します。

A

API	Application Programming Interface (アプリケーションプログラミングインタフェース)。
ASN	Abstract Syntax Notation (抽象構文記法)。
Auditing (監査)	システムに対する変更を追跡し、変更を加えたユーザーを識別すること。
auditreduce	監査証跡ファイルの監査レコードをマージおよび選択するコマンド (<code>auditreduce(1M)</code> のマニュアルページを参照)。
Authorization (承認)	Oracle Solaris OS 特権を使用してタスクやデータアクセスを実行する権限を持つユーザーを判断する方法。

B

bge	Broadcom BCM57xx デバイスの Broadcom ギガビット Ethernet ドライバ。
BSM	Basic Security Module (基本セキュリティーモジュール)。
bsmconv	BSM を有効化するコマンド (<code>bsmconv(1M)</code> のマニュアルページを参照)。
bsmunconv	BSM を無効化するコマンド (<code>bsmunconv(1M)</code> のマニュアルページを参照)。

C

CMT	Chip MultiThreading (チップマルチスレッディング)。

Compliance (適合性)	システムの構成が事前に定義されたセキュリティープロファイルに適合しているかどうかを確認すること。
Configuration (構成)	サービスプロセッサ上に保存されている論理ドメイン構成の名前。
Constraints (制約)	Logical Domains Manager にとって制約とは、特定ドメインへの割り当てが要求されたりリソースを指します。使用可能なリソースに応じて、ドメインに追加するように要求したすべてのリソースを受け取るか、まったく受け取らないかのいずれかです。
Control Domain (制御ドメイン)	Logical Domains Manager を使用してほかの論理ドメインとサービスを作成および管理する特権ドメイン。
CWQ	Control Word Queue (暗号化装置)。

D

DHCP	Dynamic Host Configuration Protocol (動的ホスト構成プロトコル)。
DIO	Direct I/O (直接 I/O)。
DMA	Direct Memory Access (ダイレクトメモリアクセス)。CPU を使用せずにメモリーとデバイス (ネットワークカードなど) との間でデータを直接転送する機能です。
DMP	Dynamic MultiPathing (動的マルチパス) (Veritas)。
Domain (ドメイン)	Logical Domain (論理ドメイン) を参照してください。
DPS	Data Plane Software (データプレーンソフトウェア)。
DR	Dynamic Reconfiguration (動的再構成)。
drd	Logical Domains Manager の Oracle Solaris OS 動的再構成デーモン (drd(1M) のマニュアルページを参照)。
DRM	Dynamic Resource Management (動的なリソース管理)。
DS	Domain Services (ドメインサービス) モジュール。
DVD	Digital Versatile Disc (デジタル多用途ディスク)。

E

EFI	Extensible Firmware Interface (拡張ファームウェアインタフェース)。

ETM Encoding Table Management (エンコーディングテーブル管理) モジュール。

F

FC_AL Fiber Channel Arbitrated Loop (ファイバチャネル調停ループ)。

FMA Fault Management Architecture (障害管理アーキテクチャー)。

fmd Oracle Solaris OS 障害管理デーモン (fmd(1M) のマニュアルページを参照)。

fmthard ハードディスクのラベルを生成するコマンド (fmthard(1M) のマニュアルページを参照)。

format ディスクのパーティション分割および保守ユーティリティ (format(1M) のマニュアルページを参照)。

G

Gb Gigabit (ギガビット)。

GLDv3 Generic LAN Driver version 3 (汎用 LAN ドライバ version 3)

Guest Domain (ゲストドメイン) I/O ドメインおよびサービスドメインのサービスを使用し、制御ドメインによって管理されます。

H

Hardening (強化) セキュリティを強化するために Oracle Solaris OS の構成を変更すること。

Hypervisor (ハイパーバイザ) オペレーティングシステムとハードウェア層の間に配置されるファームウェア層。

I

I/O 内部ディスクおよび PCIe コントローラとそれに付属するアダプタやデバイスなどの入出力装置。

I/O Domain (I/O ドメイン)	物理 I/O デバイスに対する直接の所有権と直接のアクセス権を持ち、仮想デバイスの形式でほかの論理ドメインとこれらのデバイスを共有するドメイン。
IB	InfiniBand。
IDE	Integrated Drive Electronics。
IDR	Interim Diagnostics Release。
ILOM	Integrated Lights Out Manager。オペレーティングシステムとは無関係にサーバーを管理できるようにする、ハードウェアとサポートソフトウェアから成る専用システム。
ioctl	input/output control call (I/O 制御コール)。
IPMP	Internet Protocol Network Multipathing (インターネットプロトコルネットワークマルチパス)。
K	
kaio	Kernel Asynchronous Input/Output (カーネル非同期 I/O)。
KB	Kilobyte (K バイト)。
KU	Kernel Update (カーネル更新)。
L	
LAN	Local-Area Network (ローカルエリアネットワーク)。
LDAP	Lightweight Directory Access Protocol。
LDC	Logical Domain Channel (論理ドメインチャネル)。
ldm	Logical Domains Manager ユーティリティ (ldm(1M) マニュアルページを参照)。
ldmd	Logical Domains Manager デーモン。
lofi	ループバックファイル。
Logical Domain (論理ドメイン)	リソースの個別の論理的なグループで構成される仮想マシンであり、単一のコンピュータシステム内に独自のオペレーティングシステムと識別情報を保有します。ドメインとも呼ばれます。
Logical Domains Manager	論理ドメインを作成および管理したり、リソースをドメインに割り当てたりするための CLI。

M

MAC	Media Access Control address (メディアアクセス制御アドレス) の略で、Logical Domains Manager によって自動的に割り当てることも、手動で割り当てることも可能。
MAU	Modular Arithmetic Unit (モジュラー演算ユニット)。
MB	Megabyte (M バイト)。
MD	サーバーデータベース内のマシン記述。
mem、memory	メモリー単位 – バイト単位でのデフォルトのサイズ。G バイト (G)、K バイト (K)、または M バイト (M) を指定することもできます。ゲストドメインに割り当てることができる、サーバーの仮想化されたメモリーです。
metadb	Solaris Volume Manager メタデバイス状態データベースの複製の作成と削除を行うコマンド (metadb(1M) のマニュアルページを参照)。
metaset	ディスクセットを構成するコマンド (metaset(1M) のマニュアルページを参照)。
mhd	多重ホストディスク制御操作を実行するコマンド (mhd(7i) のマニュアルページを参照)。
MIB	Management Information Base (管理情報ベース)。
Minimizing (最小化)	最低限必要な数のコア Oracle Solaris OS パッケージをインストールすること。
MMF	MultiMode Fiber (マルチモードファイバ)。
MMU	Memory Management Unit (メモリー管理ユニット)。
mpgroup	仮想ディスクフェイルオーバーのマルチパスグループ名。
mtu	Maximum Transmission Unit (最大転送単位)。

N

ndpsldcc	Netra DPS Logical Domain Channel Client。「vdpccl」も参照してください。
ndpsldcs	Netra DPS Logical Domain Channel Service。「vdpcs」も参照してください。
NIS	Network Information Service (ネットワーク情報サービス)。
NIU	Network Interface Unit (ネットワークインタフェースユニット)(オラクルの Sun SPARC Enterprise T5120 および T5220 サーバー)。
NTS	Network Terminal Server (ネットワーク端末サーバー)。

NVRAM Non-Volatile Random-Access Memory (非揮発性ランダムアクセスメモリー)。

nxge NIU 10Gb Ethernet アダプタのドライバ。

O

OID オブジェクトの識別子であり、MIB の各オブジェクトを一意に特定するシーケンス番号。

OVF Open Virtualization Format。

P

P2V Oracle VM Server for SPARC 物理から仮想への変換ツール。

PA Physical Address (物理アドレス)。

PCI Peripheral Component Interconnect バス。

PCI-X PCI 拡張バス。

PCIe PCI Express バス。

pcpu 物理 CPU。

Physical Domain (物理ドメイン) 単一 Oracle VM Server for SPARC インスタンスによって管理されるリソースの範囲。サポートされている SPARC T シリーズプラットフォームのように、物理ドメインは完全な物理システムになることがあります。または、サポートされている SPARC M シリーズプラットフォームのように、システムの全体またはシステムのサブセットになることがあります。

Physical Function (物理機能) SR-IOV 仕様に定義されたとおりに SR-IOV 機能をサポートする PCI 機能。物理機能は SR-IOV 機能構造を含み、SR-IOV 機能を管理するために使用されます。物理機能は他の PCIe デバイスと同じように検出、管理、および操作が可能な多機能の PCIe 機能です。物理機能は、すべての構成リソースを含み、PCIe デバイスを構成または制御するために使用できます。

physio 物理入出力。

PICL Platform Information and Control Library (プラットフォーム情報とコントロールライブラリ)。

picld PICL デーモン (picld(1M) のマニュアルページを参照)。

PM 仮想 CPU およびメモリーの電源管理。

praudit 監査証跡ファイルの内容を出力するコマンド (praudit(1M) のマニュアルページを参照)。

PRI Priority (優先度)。

R

RA Real Address (実アドレス)。

RAID Redundant Array of Inexpensive Disks。独立したディスクを組み合わせて 1 つの論理ユニットにすることができます。

RPC Remote Procedure Call (リモート手続き呼び出し)。

S

SAN ストレージエリアネットワーク。

SASL Simple Authentication and Security Layer。

SAX Simple API for XML パーサー。XML ドキュメントをトラバースします。SAX パーサーはイベントベースで、主にストリーミングデータに使用されます。

SCSA Sun Common SCSI Architecture。

SCSI HBA SCSI ホストバスアダプタ。

Service Domain (サービスドメイン) 仮想スイッチ、仮想コンソールコネクタ、仮想ディスクサーバーなどのデバイスをほかの論理ドメインに提供する論理ドメイン。

Service Processor (SP、サービスプロセッサ) システムコントローラ (SC) とも呼ばれる SP は、物理マシンをモニターし実行します。

SMA System Management Agent (システム管理エージェント)。

SMF Service Management Facility (サービス管理機能)。

SMI Structure of Management Information (管理情報構造)。MIB で使用する管理対象オブジェクトを定義し、グループ化します。

SNMP Simple Network Management Protocol (簡易ネットワーク管理プロトコル)。

SR-IOV Single Root I/O Virtualization (単一ルート I/O 仮想化)。

SSH Secure Shell。

ssh	Secure Shell コマンド (ssh(1) のマニュアルページを参照)。
sshd	Secure Shell デーモン (sshd(1M) のマニュアルページを参照)。
SunVTS	Sun Validation Test Suite。
svcadm	サービスインスタンスの操作 (svcadm(1M) のマニュアルページを参照)。
System Controller (SC、システム コントローラ)	「サービスプロセッサ」も参照してください。

T

TLS	Transport Layer Security。

U

UDP	User Datagram Protocol (ユーザーダイアグラムプロトコル)。
Unicast (ユニ キャスト)	1 つの送信者と 1 つの受信者の間でネットワークを介して行われる通信。
uscsi	ユーザー SCSI コマンドインタフェース (uscsi(7I) のマニュアルページを参照)。
UTP	Unshielded Twisted Pair (シールドなし・より対線)。

V

var	変数。
VBSC	Virtual Blade System Controller (仮想ブレードシステムコントローラ)。
vcc、vconscon	ゲストドメインに割り当てる特定のポート範囲の仮想コンソール端末集配信装置サービス。
vcons、vconsole	システムレベルのメッセージにアクセスするための仮想コンソール。接続は、特定のポートで制御ドメイン上の vconscon サービスに接続することによって実現します。
vcpu	Virtual Central Processing Unit (仮想中央演算処理装置)。サーバーの各コアは、仮想 CPU として表現されます。
vdc	Virtual Disk Client (仮想ディスククライアント)。

vdisk	仮想ディスクは、さまざまな種類の物理デバイス、ボリューム、またはファイルに関連付けられた総称的なブロック型デバイスです。
vdppc	Netra DPS 環境における仮想データプレーンチャネルクライアント。
vdpcs	Netra DPS 環境における仮想データプレーンチャネルサービス。
vds, vdiskserver	仮想ディスクサーバー。これを使用すると、論理ドメインに仮想ディスクをインポートできます。
vdsdev、 vdiskserverdevice	仮想ディスクサーバーデバイス。仮想ディスクサーバーによってエクスポートされます。このデバイスには、ディスク全体、ディスクのスライス、ファイル、またはディスクボリュームを指定できます。
Virtual Function (仮想機能)	物理機能に関連付けられている PCI 機能。仮想機能は、物理機能や同じ物理機能に関連付けられているほかの仮想機能と 1 つまたは複数の物理リソースを共有する軽量の PCIe 機能です。仮想機能は、その独自の動作に対してのみ構成リソースの使用が許可されます。
vldc	Virtual Logical Domain Channel Service (仮想論理ドメインチャネルサービス)。
vldcc	Virtual Logical Domain Channel Client (仮想論理ドメインチャネルクライアント)。
vnet	仮想ネットワークデバイス。仮想 Ethernet デバイスを実装し、仮想ネットワークスイッチ (vswitch) を使用するシステム内のほかの vnet デバイスと通信します。
VNIC	仮想ネットワークインタフェースカードで、これは物理ネットワークデバイスから作成し、ゾーンに割り当て可能な物理ネットワークデバイスの仮想インスタンスです。
vNTS	Virtual Network Terminal Service (仮想ネットワーク端末サービス)。
vntsd	ドメインコンソールの Oracle Solaris OS 仮想ネットワーク端末サーバーデーモン (vntsd(1M) のマニュアルページを参照)。
volfs	ボリューム管理ファイルシステム (volfs(7FS) のマニュアルページを参照)。
vsw, vswitch	仮想ネットワークデバイスを外部ネットワークに接続し、仮想ネットワークデバイス間でのパケットの切り替えも行う仮想ネットワークスイッチ。
VTOC	Volume Table Of Contents (ボリューム構成テーブル)。
VxDMP	Veritas Dynamic MultiPathing。
VxVM	Veritas Volume Manager。
X	
XFP	eXtreme Fast Path。

XML eXtensible Markup Language。

XMPP eXtensible Messaging and Presence Protocol。

Z

ZFS Zettabyte File System。

zpool ZFS ストレージプール (zpool(1M) のマニュアルページを参照)。

ZVOL ZFS ボリュームエミュレーションドライバ。

索引

あ

アクセス

ゲストドメインからファイバチャネル仮想機能, 135

移行

SSL 証明書の使用, 333

ゲストドメイン, 333

ゲストドメインと名前の変更, 334

対話型でない, 333

ドメイン, 312

移行の制限

I/O ドメイン, 62

PVLAN, 285

移行のための SSL 証明書

構成, 314

Oracle Solaris 11, 314

移行の FIPS 140-2 モード, 315

依存サイクル, 409

一覧表示

InfiniBand 仮想機能, 117

PVLAN 情報, 287

ドメインリソース, 369

マシンが読み取り可能な出力としてのリソース, 370

リソース制約, 375

インストール

DVD からの Oracle Solaris OS の, 56

ISO ファイルからの Oracle Solaris OS の, 57

JumpStart (Oracle Solaris 10) を使用した, 59

ldmp2v, 451

Oracle VM Server for SPARC MIB, 480

Oracle VM Server for SPARC MIB ソフトウェア, 481

Oracle VM Server for SPARC テンプレートユーティリティ, 434

インストールサーバーが VLAN 内に存在するときのゲストドメイン, 283

ゲストドメインへの Oracle Solaris OS の, 55

エクスポート

1 つのスライスディスクとしての ZFS ボリューム, 192

1 つのスライスディスクとしてのファイルまたはボリューム, 192

CD イメージ, 200

DVD イメージ, 200

ISO イメージ, 200

仮想ディスクとしてのファイルおよびボリューム

lofi, 194

ガイドライン, 194

仮想ディスクとしての物理ディスク, 187

仮想ディスクとしての物理ディスクスライス, 188

仮想ディスクバックエンド, 181

サービスドメインからゲストドメインへの CD または

DVD イメージ, 201

サービスドメインからゲストドメインへの ISO イメージ, 202

スライス 2, 189

ディスクスライス

間接的, 194

直接的, 194

バックエンド

比較, 193

バックエンド, サマリー, 193

ファイル, 189

複数回の CD または DVD イメージ, 202

フルディスクとしての ZFS ボリューム, 191

フルディスクとしてのファイル, 190

フルディスクとしてのファイルまたはボリューム, 189

ボリューム, 189

下位互換性, 193

エラー

CPU とメモリーアドレスのマッピングを使用したトラブルシューティング, 410

か

下位互換性

- ボリュームのエクスポート, 193

解析

- XML ベースの制御インタフェース

- Oracle VM Server for SPARC MIB, 478

解析可能なドメインリスト

- 表示, 408, 408

ガイドライン

- I/O ドメインの作成, 62

- 仮想ディスクとしてのファイルおよびボリュームのエクスポート, 194

確認

- ILOM 相互接続構成, 49

- 仮想 SCSI HBA の存在, 227

- 監査レコード, 36, 38

- ドメイン構成, 345

格納

- ZFS ファイルの使用によるディスクイメージ, 207

- ZFS ボリュームの使用によるディスクイメージ, 207

- ZFS を使用したディスクイメージ, 206

仮想機能, 85

- Ethernet, 87, 88

- Ethernet の作成, 88, 88

- Ethernet の破棄, 88, 93

- Ethernet プロパティの変更, 96

- I/O ドメインからの InfiniBand の削除, 114

- I/O ドメインからの削除, 99

- I/O ドメインからのファイバチャネルの削除, 134

- I/O ドメインの作成, 103

- I/O ドメインへの Ethernet の追加, 98

- I/O ドメインへの InfiniBand の追加, 112

- I/O ドメインへのファイバチャネルの追加, 133

- InfiniBand, 107

- InfiniBand の一覧表示, 117

- InfiniBand の作成, 108

- InfiniBand の破棄, 110

- 上の Ethernet VNIC の作成, 102

- ゲストドメインからファイバチャネルへのアクセス, 135

- デバイス固有のファイバチャネルプロパティ, 123

- の使用による I/O ドメインの Ethernet ネットワークブート, 101

- ファイバチャネル, 121

- ファイバチャネルの作成, 125

- ファイバチャネルの制限事項, 122

- ファイバチャネルの破棄, 129

- ファイバチャネルのプロパティの変更, 132

- ファイバチャネルの要件, 122, 122

- ルートドメインからの InfiniBand の削除, 117

- ルートドメインへの InfiniBand の追加, 116

- を使用した I/O ドメインの作成, 104

- 仮想機能のファイバチャネル World-Wide Name

- 割り当て, 123

仮想コンソールテーブル

- Oracle VM Server for SPARC MIB, 502

仮想スイッチ, 247

- Oracle Solaris 11 ドメインに外部接続するための構成, 267

- ドメインへの外部接続を提供するための Oracle

- Solaris 10 サービスドメイン上の構成, 268

- プライマリインタフェースとしての構成, 47

仮想ディスク, 177, 186

- 1 つのスライスディスクとしてのバックエンドエクスポート, 184

- format コマンドおよび, 205

- SCSI および, 204

- Solaris ボリュームマネージャーでの使用, 211

- VxVM での使用, 212

- ZFS での使用, 205, 213

- オプションの変更, 182

- 管理, 180

- 削除, 182

- タイムアウト, 196, 203

- タイムアウトオプションの変更, 182

- 追加, 181

- ディスク識別子, 179

- ディスク名, 179

- バックエンド, 186

- バックエンド excl オプション, 185

- バックエンド ro オプション, 184

- バックエンド slice オプション, 186

- バックエンドオプション, 184

- バックエンドのエクスポート, 181

- 表示, 183

- 物理ディスクからのエクスポート, 187

- 物理ディスクスライスからのエクスポート, 188

- フルディスクとしてのバックエンドエクスポート, 183

- ボリュームマネージャーでの使用, 210

- マルチパス, 194, 196

- マルチパスの構成, 197

仮想ディスクテーブル

- Oracle VM Server for SPARC MIB, 497
- 仮想デバイス
 - I/O, 20
 - 仮想コンソール端末集配信装置 (vcc), 21
 - 仮想スイッチ (vsw), 21
 - 仮想ディスククライアント (vdc), 21
 - 仮想ディスクサービス (vds), 21
 - 仮想ネットワーク (vnet), 21
- 仮想デバイス識別子, 258
- 仮想ドメイン情報
 - API, 413
 - virtinfo, 413
- 仮想入出力, 20
- 仮想ネットワーク, 240
 - パフォーマンスの最大化, 245, 246
- 仮想ネットワーク端末サーバーデーモン (vntsd), 21
 - 有効化, 48
- 仮想ネットワーク端末サービス (vNTS)
 - Oracle VM Server for SPARC MIB, 502
- 仮想ネットワークテーブル
 - Oracle VM Server for SPARC MIB, 500
- 仮想ネットワークデバイス, 248
 - 物理ネットワーク帯域幅の量の制御, 254
- 仮想マシン, 18
- 仮想メモリーテーブル
 - Oracle VM Server for SPARC MIB, 495
- 仮想 CPU
 - 対応する物理 CPU 番号の特定, 412
- 仮想 SCSI HBA
 - SCSI および, 236
 - 管理, 224
 - 識別子, 223
 - 存在の確認, 227
 - タイムアウト, 227, 236
 - デバイス名, 223
 - マルチパス, 230
 - マルチパスの構成, 232
 - 見え方, 229
- 環境変数
 - 設定, 485
- 監査
 - Logical Domains Manager の無効化, 38
 - Logical Domains Manager の有効化, 36
 - 無効化, 36
 - 有効化, 36
 - レコードの確認, 36, 38
- 監査レコード
 - 確認, 36, 38
- 管理
 - Oracle VM Server for SPARC MIB セキュリティー, 483
 - primary 以外のルートドメイン上の SR-IOV 仮想機能, 174
 - primary 以外のルートドメイン上の直接 I/O デバイス, 173
 - 仮想 SCSI HBA, 224
 - 仮想ディスク, 180
 - 制御ドメイン上の物理リソース, 355
 - ドメイン構成, 381
 - リソースグループ, 364
- 管理特権の委任
 - 権利プロファイル, 25
- 起動
 - Oracle VM Server for SPARC MIB を使用したドメイン, 515
 - ゲストドメイン, 51
 - ドメイン, 516
- クローニング
 - ブートディスクイメージ, 209
- 計画
 - Ethernet SR-IOV, 87
 - PCIe SR-IOV 仮想機能の, 85
 - 直接 I/O (DIO), 152
 - 直接 I/O (Direct I/O, DIO), 152
- ゲストコンソール
 - ネットワークを介した接続, 401
- ゲストドメイン, 19
 - 移行, 333
 - 移行と名前の変更, 334
 - 起動, 51
 - 作成, 51
 - すべて削除, 428
- 結合
 - 単一グループへのコンソール, 402
- 減少
 - 制御ドメイン上のメモリー, 358
- 権利プロファイル
 - 割り当て, 25, 27
- コア全体の制約
 - 適用, 341
- コアの最大数の制約
 - 適用, 341

更新

/etc/system ファイル, 400
PVLAN, 287

構成

CPU コア全体を使用した既存のドメイン, 347
CPU コア全体を使用した制御ドメイン, 348
CPU コア全体を使用したドメイン, 346
ldmp2v, 451
NAT およびルーティングのための仮想スイッチおよびサービスドメイン, 265
Oracle Solaris 10 システム上の NAT, 267
Oracle Solaris 11 システム上の NAT, 265
Oracle Solaris 11 ドメインに外部接続するための仮想スイッチ, 267
Oracle VM Server for SPARC MIB, 480
Oracle VM Server for SPARC MIB ソフトウェア, 481
Oracle VM Server for SPARC P2V ツール, 453
Oracle VM Server for SPARC 環境での IPMP, 269
PMP グループへの仮想ネットワークデバイス, 269, 271
移行のための SSL 証明書, 314
 Oracle Solaris 11, 314
仮想 SCSI HBA マルチパス, 232
仮想ディスクマルチパス, 197
サービスドメインでの IPMP, 272
サービスドメイン内の ZFS プール, 206
ジャンボフレーム, 299
制御ドメイン, 43
ドメインの依存関係, 406
ドメインへの外部接続を提供するための Oracle Solaris 10 サービスドメイン上の仮想スイッチ, 268
ハードパーティションを使用したシステム, 344
パフォーマンスレジスタアクセス, 376
ブートの選択, 22
物理リンクステータスの更新, 275
プライマリインタフェースとしての仮想スイッチ, 47

構成解除

障害のあるハードウェアリソース, 392
コヒーレンシリンクのスケーリング, 469
コマンド行インタフェース, 20
コンソール
 単一グループへの結合, 402

ロギング, 39
コンソールグループ
 使用, 401

さ

サーバー

 電源再投入の実行, 404
サービスドメイン, 19, 20
 ZFS プール, 206
サービスプロセッサ (SP)
 出荷時デフォルト構成の復元, 430
 での Oracle VM Server for SPARC の使用, 405
 物理マシンのモニタリングと実行, 18
最大化
 仮想ネットワークパフォーマンス, 245, 246
再バインド
 CPU コア全体を使用したシステム, 351
削除, 99
 参照 破棄
 I/O ドメインからの Ethernet 仮想機能, 99
 I/O ドメインからのファイバチャネル仮想機能, 134
 Oracle VM Server for SPARC MIB ソフトウェア, 481, 482
 physical-bindings 制約, 353
 PVLAN, 287
 仮想ディスク, 182
 すべてのゲストドメイン, 428
 すべてのドメイン構成, 428
 ドメインからのメモリー, 357
 物理的にバインドされていないリソース, 354
 ルートドメインからの InfiniBand 仮想機能, 117

作成

CPU コア全体を使用したドメイン, 346
Ethernet 仮想機能, 88, 88
Ethernet 仮想機能上の VNIC, 102
I/O ドメインの Ethernet 仮想機能, 103
InfiniBand 仮想機能, 108
PCIe バス全体からルートドメインを, 66
PVLAN, 287
snmpv3 ユーザー, 483
ゲストドメイン, 51
構成解除されたシステムのディスクイメージスナップショット, 209
制御ドメインのデフォルトサービス, 42

- ディスクイメージスナップショット, 207, 208
- ファイバチャネル仮想機能, 125
- 役割, 27
- 識別
 - InfiniBand 機能, 119
- システム管理エージェント
 - Oracle VM Server for SPARC MIB, 477
- システムコントローラ 参照 サービスプロセッサ (SP)
- 自動保存構成ディレクトリ
 - 復元, 426
 - 保存, 426
- ジャンボフレーム
 - 構成, 299
 - ジャンボフレームに対応していないバージョンの Oracle Solaris 10 vnet および vsw ドライバとの互換性, 303
- 縮小
 - 制御ドメインの CPU およびメモリーリソース, 45
- 受信
 - Oracle VM Server for SPARC MIB トラップ, 509
- 出荷時デフォルト構成
 - サービスプロセッサからの復元, 430
 - 復元, 429
- 取得
 - Oracle VM Server for SPARC MIB オブジェクト
 - snmpget, 486
 - Oracle VM Server for SPARC MIB オブジェクト値
 - snmptable, 487
 - snmpwalk, 486
 - Oracle VM Server for SPARC MIB 情報, 488
 - ドメインの移行ステータス, 334
- 使用
 - VLAN, 281
 - リンクベースの IPMP, 274
- 障害管理アーキテクチャ (FMA)
 - ブラックリスト登録, 391
- 障害と復旧の情報
 - 提供, 478
- 障害のあるハードウェアリソース
 - 構成解除, 392
 - ブラックリスト登録, 392
 - を含むドメインの復旧, 393
- 障害ポリシー
 - ドメインの依存関係に対する構成, 407
- 承認
 - ldm サブコマンド, 29
- 消費電力データ
 - 表示, 469
- スライス 2
 - エクスポート, 189
- 制御
 - 回復モード, 396
- 制御ドメイン, 19
 - 構成, 43
 - メモリーの減少, 358
 - メモリーの再構成, 358
 - リブート, 46, 404
- 制御ドメインのデフォルトサービス
 - 作成, 42
- 制御ドメインの CPU およびメモリーリソース
 - 縮小, 45
- 制限
 - PVLAN, 285
 - 物理ネットワーク帯域幅, 255
 - リソースグループ, 365
- 制限事項
 - Ethernet SR-IOV, 87
 - primary 以外のルートドメイン, 169
 - SR-IOV, 80
 - 直接 I/O, 151
 - ファイバチャネル仮想機能, 122
- 接続
 - ネットワークを介したゲストコンソールへの, 401
- 設定
 - 環境変数, 485
 - 電力の制限, 469
 - ドメインのメモリーサイズ, 363
 - 物理ネットワーク帯域幅の制限, 255
- 送信
 - Oracle VM Server for SPARC MIB トラップ, 508
- た
 - 回避された I/O リソース, 397
 - タイムアウトオプション
 - 仮想ディスク, 182
 - 対話型でないドメインの移行, 333
 - 遅延再構成, 338, 358

- 調整
 - 割り込み制限, 421
- 直接 I/O (DIO)
 - primary 以外のルートドメイン上のデバイスの管理, 173
 - 計画, 152
 - 制限事項, 151
 - 要件, 150
- 追加
 - I/O ドメインからの InfiniBand 仮想機能, 114
 - I/O ドメインへの Ethernet 仮想機能, 98
 - I/O ドメインへの InfiniBand 仮想機能, 112
 - I/O ドメインへのファイバチャネル仮想機能, 133
 - 仮想ディスク, 181
 - 単位が調整されていないメモリー, 360
 - ドメインへのメモリー, 357
 - ルートドメインへの InfiniBand 仮想機能, 116
- 停止
 - Oracle VM Server for SPARC MIB を使用したドメイン, 515
 - 高負荷のドメイン, 402
 - ドメイン, 517
- ディスクイメージ
 - ZFS ファイルの使用による格納, 207
 - ZFS ボリュームの使用による格納, 207
 - ZFS を使用した格納, 206
 - 構成解除されたシステムのスナップショットの作成, 209
 - スナップショットの作成, 207, 208
- ディスクスライス 参照 物理ディスクスライス
- テーブル 参照 Oracle VM Server for SPARC MIB
- テーブル
- デーモン
 - drd, 338
 - ldmd, 20
 - vntsd, 21
- 適用
 - コア全体の制約, 341
 - コアの最大数の制約, 341
- デバイス固有のプロパティー
 - Ethernet SR-IOV, 101
- 電源管理 (PM), 468, 468, 468, 469, 469, 469
 - CPU, 351
 - 可観測性モジュール
 - 有効化, 470
 - 機能, 468
 - 使用, 365, 467
- 電源再投入
 - サーバー上での実行, 404
- 電力の制限, 469
- 問い合わせ
 - Oracle VM Server for SPARC MIB, 485
- 動的再構成デーモン (drd), 338
- 動的再構成 (DR), 338, 358
 - CPU, 343, 349
 - 部分的なメモリー要求, 357
 - メモリー, 356
- 動的なパス選択, 198
- 動的なリソース管理, 343, 351
 - 使用, 366
- 特定
 - ドメイン構成, 411
- ドメイン
 - Oracle VM Server for SPARC MIB を使用したモニタリング, 484
 - 移行, 312
 - 依存関係, 406
 - 依存関係の構成, 406
 - 依存サイクル, 409
 - 起動, 516
 - クローンの使用によるプロビジョニング, 208
 - 高負荷の停止, 402
 - サービス, 20
 - 縮退としてマーク, 397
 - 定義, 16
 - 停止, 517
 - ドメインの依存関係に対する障害ポリシーの構成, 407
 - の種類, 19, 19
 - のタイプ, 19, 19
 - 無効化, 428
 - 役割, 19, 19
- ドメイン移行の制限, 318
- ドメイン構成
 - XML ファイルからの復元 `ldm add-domain`, 388
 - XML ファイルからの復元 `ldm init-system`, 387
 - 永続的, 22
 - 確認, 345
 - 管理, 381
 - 自動回復ポリシー, 384
 - 自動保存を使用した復元, 382
 - 縮退, 396

- すべて削除, 428
- 特定, 411
- の自動回復ポリシー, 383
- 復元, 382, 387
- 保存, 382, 385
- ドメイン構成の回復モード
 - 制御, 396
- ドメイン構成の自動回復ポリシー, 383, 384
- ドメイン構成の復旧モード, 391
- ドメインコンソール
 - へのアクセスの制御, 30
- ドメインスカラー変数
 - Oracle VM Server for SPARC MIB, 492
- ドメインの移行, 321
 - CPU の要件, 323
 - NIU ハイブリッド I/O の要件, 328
 - OpenBoot PROM からまたはカーネルデバッグで, 329
 - PCIe エンドポイントデバイスの要件, 327, 331
 - SR-IOV 仮想機能の要件, 328, 331
 - アクティブ, 322
 - アクティブドメインの遅延再構成, 329
 - アクティブなドメインで電源管理のエラスティックポリシーが有効になっている場合, 329
 - 暗号化ユニットの要件, 328
 - 仮想 I/O デバイスの要件, 326, 330
 - 失敗からの復旧, 332
 - 失敗メッセージ, 334
 - 進行中の取り消し, 332
 - 進捗のモニタリング, 331
 - ステータスの取得, 334
 - セキュリティ, 313
 - 操作, 312
 - ソフトウェアの互換性, 313
 - 対話型でない, 333
 - バインドされたドメインまたはアクティブでないドメイン, 330
 - 非対話式的実行, 321
 - 物理 I/O デバイスの要件, 326
 - ほかのドメインの操作, 329
 - メモリの要件, 325
 - 予行演習の実行, 321
- ドメインリスト
 - 解析可能な, 408
- ドメインリソース
 - 一覧表示, 369

- ドメインリソースプール
 - Oracle VM Server for SPARC MIB, 492
- トラップ 参照 Oracle VM Server for SPARC MIB
- トラップ
- トラブルシューティング
 - CPU とメモリアドレスのマッピング, 410
 - Oracle VM Server for SPARC, 23

な

- ネットワークインタフェース名, 258
 - 検索, 260
- ネットワーク構成
 - Ethernet SR-IOV, 100
- ネットワークデバイス
 - 使用, 264
 - ネットワーク帯域幅の制限, 設定, 254
- ネットワークデバイス構成
 - 表示, 251
- ネットワークデバイス統計情報
 - 表示, 251
- ネットワークブート
 - Ethernet SR-IOV 仮想機能の使用による I/O ドメイン, 101

は

- ハードウェアエラー
 - トラブルシューティング, 391
- ハードパーティション
 - を使用したシステムの構成, 344
- ハイパーバイザ
 - Logical Domains Manager と, 16
 - 定義, 16
- 破棄, 88
 - 参照 削除
 - Ethernet 仮想機能, 88, 93
 - InfiniBand 仮想機能, 110
 - ファイバチャネル仮想機能, 129
- バスの割り当て
 - 静的, 64
 - 動的, 65
- バックエンド, 193
 - 参照 仮想ディスクバックエンドのエクスポート
- パフォーマンス
 - 仮想ネットワークの最大化, 245, 246

- 仮想ネットワークを最大にするための要件, 245
 - パフォーマンスレジスタアクセス
 - 設定, 376
 - 判定
 - ネットワークデバイスの GLDv3 コンプライアンス (Oracle Solaris 10), 264
 - 表示
 - 解析可能なドメインリスト, 408
 - 消費電力データ, 469
 - ネットワークデバイス構成, 251
 - ネットワークデバイス統計情報, 251
 - プロセッサの消費電力データ, 472
 - ブート
 - SCSI DVD デバイス, 235
 - ブートディスクイメージ
 - クローニング, 209
 - 復元
 - Logical Domains 制約データベース, 427
 - サービスプロセッサからの出荷時デフォルト構成, 430
 - 自動保存構成ディレクトリ, 426
 - 出荷時デフォルト構成, 429
 - ドメイン構成, 382, 387
 - XML ファイルからの `ldm add-domain`, 388
 - XML ファイルからの `ldm init-system`, 387
 - 復旧
 - 障害のあるハードウェアリソースを含むドメイン, 393
 - ドメイン移行の失敗からの, 332
 - 見つからないハードウェアリソースを含むドメイン, 393
 - 物理ディスク
 - 仮想ディスクとしてのエクスポート, 187
 - 物理ディスクスライス, 188
 - 仮想ディスクとしてのエクスポート, 188
 - 物理ディスク LUN, 186
 - 物理的にバインドされていないリソース
 - 削除, 354
 - 物理デバイス, 19, 20
 - 物理ネットワーク帯域幅
 - 仮想ネットワークデバイスによる使用の制御, 254
 - 制限, 255
 - 制限の設定, 255
 - 物理マシン, 18
 - 物理リソース
 - 管理上の制約, 355
 - 制御ドメイン上の管理, 355
 - ドメインへの割り当て, 351
 - 物理リンクステータスの更新
 - 構成, 275
 - 物理 CPU 番号
 - 対応する仮想 CPU の特定, 412
 - プライベート VLAN (PVLAN)
 - 使用, 284
 - ブラックリスト登録
 - 障害管理アーキテクチャー (FMA), 391
 - 障害のあるハードウェアリソース, 392
 - ブレーク
 - Oracle Solaris OS, 404
 - プロセッサの消費電力データ
 - 表示, 472
 - プロパティ
 - Ethernet SR-IOV デバイス固有の, 88
 - ファイバチャネル仮想機能デバイス固有のプロパティ, 123
 - プロビジョニング
 - クローンの使用によるドメイン, 208
 - 変更
 - Ethernet SR-IOV 仮想機能のプロパティ, 96
 - PCIe ハードウェアへの変更, 156
 - 仮想ディスクオプション, 182
 - 仮想ディスクのタイムアウトオプション, 182
 - ドメイン構成の自動回復ポリシー, 384
 - ファイバチャネル仮想機能のプロパティ, 132
 - ポート VLAN ID (PID), 280
 - 保存
 - Logical Domains 制約データベース, 427
 - 自動保存構成ディレクトリ, 426
 - ドメイン構成, 382, 385
 - ボリュームマネージャー
 - 仮想ディスクでの使用, 210
- ま**
- マシンが読み取り可能な出力
 - リソースの一覧表示, 370
 - マスタートメイン
 - 割り当て, 407
 - マルチパス 参照 仮想 SCSI HBA マルチパス 参照
 - 仮想ディスクマルチパス
 - 見つからないハードウェアリソースを含むドメインの復旧, 393
 - 無効化

- Logical Domains Manager, 429
- Logical Domains Manager の監査, 38
- NIU ハイブリッド I/O, 296
- 監査, 36
- ドメイン, 428
- メモリー
 - アクティブでないドメインの配置, 360
 - 制御ドメイン上の減少, 358
 - 単位が調整されていない追加, 360
 - ドメインからの削除, 357
 - ドメインのサイズの設定, 363
 - ドメインへの追加, 357
 - 配置, 359
 - アクティブなドメインの, 359
 - バインドされているドメインの配置, 359
 - マッピング, 411
- メモリーサイズの要件, 55
- メモリーの再構成
 - 制御ドメイン, 358
- メモリーの動的再構成
 - アクティブなドメイン上の操作, 361
 - バインドされているドメイン上の操作, 362
 - 部分的な要求, 357
- メモリーの動的再構成 (DR), 356
- メモリー DR 参照 メモリーの動的再構成 (DR)
- モニタリング
 - Oracle VM Server for SPARC MIB を使用したドメイン, 484
- や
- 役割
 - 作成, 27
 - ドメイン, 19
 - ユーザーへの割り当て, 27
 - 割り当て, 25
- 有効化
 - I/O 仮想化, 84
 - ILOM 相互接続サービス, 50
 - Logical Domains Manager デーモン, 426
 - Logical Domains Manager の監査, 36
 - NIU ハイブリッド I/O, 296
 - PCIe バスの I/O 仮想化, 171
 - 仮想ネットワーク端末サーバーデーモン (vntsd), 48
 - 監査, 36
 - 電源管理可観測性モジュール, 470
- ユニバーサル固有識別子 (UUID), 413
- 要件
 - Ethernet SR-IOV, 87
 - InfiniBand SR-IOV, 108
 - PVLAN, 285
 - SR-IOV, 76
 - 仮想ネットワークパフォーマンスを最大にするための, 245
 - 静的 SR-IOV の, 82
 - 直接 I/O, 150
 - 動的 SR-IOV の, 83, 83
 - ファイバチャネル仮想機能, 122, 122
 - リソースグループ, 365
- ら
- リソース, 17
 - 参照 仮想デバイス
 - 出力のフラグ定義, 370
 - 定義, 17
 - 割り当て, 339
- リソース管理
 - 動的, 343
- リソースグループ
 - 管理, 364
 - 制限, 365
 - 要件, 365
- リソース構成, 22
- リソース制約
 - 一覧表示, 375
- リソースの割り当て, 339
- リブート
 - CPU コア全体を使用したシステム, 351
 - 制御ドメイン, 46
 - ルートドメイン, 144, 154
- 利用統計情報, 371
- リンクアグリゲーション
 - 仮想スイッチでの使用, 296
- リンクベースの IPMP
 - 使用, 274
- ルーティング
 - 仮想スイッチおよびサービスドメインの構成, 265
- ルートドメイン, 19, 63
 - PCIe バスの割り当てによる作成, 63
 - 作成, 66
 - リブート, 144, 154

ロード

Oracle Solaris SNMP エージェント への Oracle VM Server for SPARC MIB モジュール, 482
Oracle VM Server for SPARC MIB モジュール, 481

論理ドメインチャネル (LDC), 18, 414
inter-vnet, 249

わ

割り当て

CPU リソース, 340
I/O ドメインへのエンドポイントデバイス, 147
MAC アドレス, 261
自動, 261
手動, 261
Oracle Solaris 10 ゲストドメイン内の VLAN, 282
Oracle Solaris 10 サービスドメイン内の VLAN, 281
Oracle Solaris 11 ゲストドメイン内の VLAN, 282
Oracle Solaris 11 サービスドメイン内の VLAN, 281
PCIe エンドポイントデバイス, 66
VLAN, 281
権利プロファイル, 25, 27
ドメインへの物理リソース, 351
ファイバチャネル仮想機能の World-Wide Name, 123
マスタードメイン, 407
役割, 25
ユーザーへの役割, 27
リソース, 339
ルートドメインへの PCIe バス, 63

割り込み制限

調整, 421

C

cancel-reconf サブコマンド, 339

CD イメージ

エクスポート, 200

CD または DVD イメージ

サービスドメインからゲストドメインへのエクスポート, 201

複数回エクスポート, 202

CLI 参照 コマンド行インタフェース

CPU

CPU の動的なリソース管理, 351

CPU DR, 343, 349

CPU クロックサイクルのスキップ, 468

CPU コア全体

を使用した既存のドメインの構成, 347

を使用したシステムの再バインド, 351

を使用したシステムのリブート, 351

を使用した制御ドメインの構成, 348

を使用したドメインの構成, 346

を使用したドメインの作成, 346

CPU コアの無効化, 468

CPU とメモリアドレスのマッピング

トラブルシューティング, 410

CPU の電源管理, 351

CPU の動的な電圧および周波数スケーリング (DVFS), 468

CPU の割り当て, 340

CPU マッピング, 410

CPU リソース

割り当て, 340

D

dmp2v

バックエンドデバイス, 450

DR 参照 動的再構成 (DR)

DVD イメージ

エクスポート, 200

E

/etc/system ファイル

更新, 400

Ethernet SR-IOV

計画, 87

制限事項, 87

デバイス固有のプロパティ, 88, 101

ネットワーク構成, 100

要件, 87

Ethernet SR-IOV 仮想機能の使用による I/O ドメインのブート, 101

F

FMA 参照 障害管理アーキテクチャー (FMA)
format
仮想ディスク, 205

G

GLD コンプライアンス (Oracle Solaris 10)
ネットワークデバイスの, 264

I

I/O 仮想化
PCIe バスの有効化, 171
有効化, 84
I/O ドメイン, 61, 73, 147
PCIe SR-IOV 仮想機能の使用, 73
PCIe バス, 61
SR-IOV 仮想機能の割り当てによる作成, 73
SR-IOV 仮想機能の割り当てによるブート, 101
移行の制限, 62
エンドポイントデバイスの割り当てによる作成, 147
作成のガイドライン, 62
I/O リソース
退避としてマーク, 397
ILOM 相互接続構成
verificating, 49
ILOM 相互接続サービス
有効化, 50
InfiniBand SR-IOV
要件, 108
inter-vnet LDC チャネル, 249
inter-vnet リンク
PVLAN, 284
IPMP
Oracle VM Server for SPARC 環境での構成,
269
グループへの仮想ネットワークデバイスの構成,
269, 271
サービスドメインでの構成, 272
ISO イメージ
エクスポート, 200
サービスドメインからゲストドメインへのエクスポート,
202

J

JumpStart
を使用したゲストドメインへの Oracle Solaris 10
のインストール, 59

L

LDC 参照 論理ドメインチャネル (LDC)
ldmd 参照 Logical Domains Manager デーモン
ldmp2v(1M) コマンド, 448
ldmp2v
Oracle VM Server for SPARC P2V ツール, 22
Oracle VM Server for SPARC P2V 変換ツール,
447
インストール, 451
構成, 451
収集フェーズ, 448, 454
準備フェーズ, 449, 456
制限事項, 452
前提条件, 452
変換フェーズ, 450, 458
lofi
仮想ディスクとしてのファイルおよびボリュームのエクスポート, 194
Logical Domains Manager, 16, 18
Oracle VM Server for SPARC MIB と, 478
XML スキーマの使用, 525
検出メカニズム, 521
デーモン (ldmd), 20
無効化, 429
Logical Domains Manager デーモン
有効化, 426
Logical Domains 制約データベース
復元, 427
保存, 427

M

MAC アドレス
自動割り当て, 261
自動割り当てアルゴリズム, 263
重複の検出, 263
手動割り当て, 261
ドメインに割り当てられた, 262
割り当て, 261

N**NAT**

- Oracle Solaris 10 システム上の構成, 267
- Oracle Solaris 11 システム上の構成, 265
- 仮想スイッチおよびサービスドメインの構成, 265

NIU ハイブリッド I/O

- 使用, 291
- 無効化, 296
- 有効化, 296

O

- Oracle Solaris 10 ネットワーク, 243
- Oracle Solaris 11 ネットワーク, 240
- Oracle Solaris 11 のネットワーク固有の機能の相違点, 304

Oracle Solaris OS

- Oracle VM Server for SPARC の操作, 403
- ゲストドメインへのインストール, 55
 - DVD からの, 56
 - ISO ファイルからの, 57
- ネットワークインタフェース名 (Oracle Solaris 11)
 - 検索, 259
 - ブレイク, 404

Oracle Solaris SNMP エージェント

- への Oracle VM Server for SPARC MIB モジュールのロード, 482

Oracle VM Server for SPARC MIB, 23, 475, 475

- Logical Domains Manager と, 478
- Oracle VM Server for SPARC の, 23
- XML ベースの制御インタフェース
 - 解析, 478
- インストール, 480
- オブジェクトツリー, 479
- 概要, 475
- 仮想コンソールテーブル, 502
- 仮想ディスクテーブル, 497
- 仮想ネットワーク端末サービス (vNTS), 502
- 仮想ネットワークテーブル, 500
- 仮想メモリーテーブル, 495
- 関連の製品および機能, 476
- 構成, 480
- システム管理エージェント, 477
- ソフトウェアコンポーネント, 476
- 問い合わせ, 485

ドメインスカラー変数, 492

ドメインの起動, 515

ドメインの停止, 515

ドメインリソースプール, 492

Oracle VM Server for SPARC MIB オブジェクト取得

snmpget, 486

Oracle VM Server for SPARC MIB オブジェクト値

取得

snmptable, 487

snmpwalk, 486

Oracle VM Server for SPARC MIB セキュリティー管理, 483**Oracle VM Server for SPARC MIB ソフトウェアインストール, 481**

構成, 481

削除, 481, 482

Oracle VM Server for SPARC MIB テーブル

CMI テーブル (ldomCMITable), 506

CMI リソースプールのスカラー変数, 494

CPU リソースプールのスカラー変数, 492

I/O バステーブル (ldomIOBusTable), 506

I/O バスリソースプールのスカラー変数, 493

暗号化ユニットテーブル (ldomCryptoTable), 505

暗号化リソースプールのスカラー変数, 493

仮想 CPU テーブル (ldomVcpuTable), 494

仮想コンソール関係テーブル

(ldomVconsVccRelTable), 504

仮想コンソールグループテーブル

(ldomVconsTable), 503

仮想コンソール端末集配信装置テーブル

(ldomVccTable), 503

仮想スイッチサービスデバイステーブル

(ldomVswTable), 501

仮想ディスクサービステーブル (ldomVdsTable), 497

仮想ディスクサービスデバイステーブル

(ldomVdsdevTable), 497

仮想ディスクテーブル (ldomVdiskTable), 498

仮想ネットワークデバイステーブル

(ldomVnetTable), 502

仮想メモリーテーブル (ldomVmemTable), 496

仮想メモリーの物理バインディングテーブル

(ldomVmemPhysBindTable), 496

環境変数テーブル (ldomEnvVarsTable), 490
 コアテーブル (ldomCoreTable), 507
 サービスプロセッサ構成テーブル
 (ldomSPConfigTable), 492
 ドメインテーブル (ldomTable), 488
 ドメインバージョン情報のスカラー変数, 507
 ドメインポリシーテーブル (ldomPolicyTable), 491
 Oracle VM Server for SPARC MIB トラップ, 510
 CMI リソースの変更 (ldomCMIChange), 515
 仮想 CPU の変更 (ldomVCpuChange), 511
 仮想コンソールグループの変更
 (ldomVconsChange), 515
 仮想コンソール端末集配信装置の変更
 (ldomVccChange), 514
 仮想スイッチの変更 (ldomVswChange), 513
 仮想ディスクサービスの変更 (ldomVdsChange),
 512
 仮想ディスクの変更 (ldomVdiskChange), 513
 仮想ネットワークの変更 (ldomVnetChange), 514
 仮想メモリーの変更 (ldomVMemChange), 511
 受信, 509
 送信, 508
 ドメイン状態の変更 (ldomStateChange), 511
 ドメインの削除 (ldomDestroy), 510
 ドメインの作成 (ldomCreate), 510
 Oracle VM Server for SPARC MIB モジュール
 Oracle Solaris SNMP エージェント へのロード,
 482
 ロード, 481
 Oracle VM Server for SPARC MIB モジュールト
 ラップ
 受信, 508
 送信, 508
 Oracle VM Server for SPARC
 サービスプロセッサでの使用, 405
 トラブルシューティング, 23
 Oracle VM Server for SPARC P2V ツール
 ldmp2v, 22, 447
 構成, 453
 制限事項, 452
 Oracle VM Server for SPARC 管理情報ベース
 (Management Information Base, MIB), 475 参
 照 Oracle VM Server for SPARC MIB
 Oracle VM Server for SPARC テンプレートユーティ
 リティー

インストール, 434

P

PCIe SR-IOV 仮想機能 参照 仮想機能
 の計画, 85
 PCIe バス, 61
 I/O 仮想化の有効化, 84
 ハードウェアの変更, 156
 physical-bindings 制約
 削除, 353
 primary 以外のルートドメイン, 167
 PCIe SR-IOV 仮想機能の割り当て, 167
 PCIe エンドポイントデバイスの割り当て, 167
 SR-IOV 仮想機能の管理, 174
 概要, 167
 制限事項, 169, 169
 直接 I/O デバイスの管理, 173
 primaryドメイン, 19
 PVLAN
 inter-vnet リンク, 284
 移行の制限, 285
 更新, 287
 削除, 287
 作成, 287
 情報の一覧表示, 287
 制限, 285
 要件, 285

R

ro オプション
 仮想ディスクバックエンド, 184

S

SCSI および仮想ディスク, 204
 SCSI および 仮想 SCSI HBA, 236
 slice オプション
 仮想ディスクバックエンド, 186
 SNMP トラップ
 使用, 508
 提供, 478
 snmpget
 取得

- Oracle VM Server for SPARC MIB オブジェクト, 486
- snmptable
 - Oracle VM Server for SPARC MIB オブジェクト値の取得, 487
- snmpv3 ユーザー
 - 作成, 483
- snmpwalk
 - Oracle VM Server for SPARC MIB オブジェクト値の取得, 486
- Solaris Power Aware Dispatcher (PAD), 469
- Solaris ボリュームマネージャー
 - 仮想ディスクでの使用, 211
 - 使用, 213
- SR-IOV, 73
 - Ethernet デバイス固有のプロパティ, 88
 - 機能タイプ, 74
 - 制限事項, 80
 - 静的, 81
 - 静的の要件, 82
 - 動的, 83
 - 動的の要件, 83, 83
 - 要件, 76
- SR-IOV 仮想機能 参照 仮想機能
- SSL 証明書
 - 移行, 333
- SUNwldm パッケージ, 20
- V**
- virtinfo
 - 仮想ドメイン情報, 413
- VLAN ID (VID), 280
- VLAN
 - Oracle Solaris 10 ゲストドメイン内での割り当て, 282
 - Oracle Solaris 10 サービスドメイン内での割り当て, 281
 - Oracle Solaris 11 ゲストドメイン内での割り当て, 282
 - Oracle Solaris 11 サービスドメイン内での割り当て, 281
 - 使用, 281
 - 割り当て, 281
- VLAN タグ付け
 - 使用, 279
- VNIC
 - SR-IOV 仮想機能の作成, 102
- VxVM
 - 仮想ディスクでの使用, 212
 - 使用, 214
- X**
- XML
 - Logical Domains Manager のリソースとプロパティ, 540
 - <LDM_event> メッセージ, 533
 - アクション, Logical Domains Manager, 538
 - 応答メッセージ, 530
 - オブジェクトの応答, 532
 - コマンドの応答, 531
 - スキーマ, 559
 - 全体の応答, 531
 - ドメインの移行, 558
 - 要求メッセージ, 527
 - 要求メッセージと応答メッセージ, 527
- XML イベント
 - 進捗, 535
 - すべて, 537
 - タイプ, 534
 - 登録および登録解除, 532
 - ドメイン, 534
 - ハードウェア, 535
 - メッセージ, 532
 - リソース, 536
- XML スキーマ, 559
 - Logical Domains Manager での使用, 525
- XML タグ
 - <cmd>, 528
 - <data>, 529
 - <LDM_interface>, 528
- XML トランスポートフレーム, 525
- XML プロトコル, 526
- XML ベースの制御インタフェース
 - Oracle VM Server for SPARC MIB 解析, 478, 478
- XML リソース
 - console, 557
 - cpu, 542
 - disk, 546

- ldom_info, 540
- mau, 544
- memory, 545
- network, 549
- physio_device, 551
- policy, 555
- spconfig, 554
- var, 550
- vcc, 550
- vdpc, 556
- vdpcs, 556
- vds_volume, 546
- vds, 545
- vsw, 547

XMPP

- サーバー, 526
- ローカル接続, 526

Z

ZFS

- 仮想ディスクおよび, 205
- 仮想ディスクでの使用, 213
- を使用したディスクイメージの格納, 206

ZFS ファイル

- の使用によるディスクイメージの格納, 207

ZFS プール

- サービスドメイン内の構成, 206

ZFS ボリューム

- 1 つのスライスディスクとしてのエクスポート, 192
- 仮想ディスクバックエンドの複数回エクスポート, 181
- の使用によるディスクイメージの格納, 207
- フルディスクとしてのエクスポート, 191

