

Oracle® VM Server for SPARC 3.0 管理ガイド

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

目次

はじめに	15
パート I Oracle VM Server for SPARC 3.0 ソフトウェア	19
1 Oracle VM Server for SPARC ソフトウェアの概要	21
ハイパーバイザと Logical Domains	22
Logical Domains Manager	24
ドメインの役割	25
コマンド行インタフェース	26
仮想入出力	26
リソースの構成	27
持続的な構成	27
Oracle VM Server for SPARC 物理から仮想への変換ツール	28
Oracle VM Server for SPARC Configuration Assistant	28
Oracle VM Server for SPARC 管理情報ベース	29
2 ソフトウェアのインストールおよび有効化	31
必須の Oracle VM Server for SPARC ソフトウェアコンポーネント	31
新しいシステムへの Oracle VM Server for SPARC ソフトウェアのインストール	32
Oracle Solaris OS の更新	33
システムファームウェアのアップグレード	33
Logical Domains Manager のダウンロード	34
Logical Domains Manager のインストール	35
Logical Domains Manager デーモンの有効化	37
Oracle VM Server for SPARC をすでに使用しているシステムのアップグレード	38
Oracle Solaris OS のアップグレード	39
Logical Domains Manager およびシステムファームウェアのアップグレード	40

Oracle VM Server for SPARC 3.0 ソフトウェアへのアップグレード	41
出荷時デフォルト構成と Logical Domains の無効化	44
▼すべてのゲストドメインを削除する方法	44
▼すべての論理ドメイン構成を削除する方法	44
▼出荷時デフォルト構成を復元する方法	45
▼Logical Domains Manager を無効にする方法	45
▼Logical Domains Manager を削除する方法	45
▼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	46
3 Oracle VM Server for SPARC のセキュリティー	47
RBAC の使用による論理ドメインの管理の委任	47
権利プロファイルと役割の使用	48
Logical Domains Manager プロファイルの内容	51
RBAC の使用によるドメインコンソールへのアクセスの制御	51
▼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	53
▼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	54
▼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	56
▼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 ...	58
監査の有効化と使用	59
▼監査を有効にする方法	59
▼監査を無効にする方法	61
▼監査レコードを確認する方法	61
▼監査ログをローテーションする方法	62
4 サービスおよび制御ドメインの設定	63
出力メッセージ	63
デフォルトのサービスの作成	64
▼デフォルトのサービスを作成する方法	64
制御ドメインの初期構成	65
▼制御ドメインを設定する方法	66
Logical Domains を使用するためのリポート	67
▼リポートする方法	67
制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効 化	67

▼ 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	68
仮想ネットワーク端末サーバーデーモンの有効化	69
▼ 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	69
5 ゲストドメインの設定	71
ゲストドメインの作成と起動	71
▼ ゲストドメインを作成および起動する方法	71
ゲストドメインへの Oracle Solaris OS のインストール	74
▼ 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	75
▼ 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	76
▼ Oracle Solaris 10 ゲストドメインで Oracle Solaris JumpStart 機能を使用する方法 ...	78
6 I/O ドメインの設定	81
I/O ドメインの概要	81
I/O ドメインを作成するための一般的なガイドライン	82
PCIe バスの割り当て	83
▼ PCIe バスの割り当てによって I/O ドメインを作成する方法	84
PCIe エンドポイントデバイスの割り当て	89
直接 I/O のハードウェア要件とソフトウェア要件	91
直接 I/O 機能の現在の制限事項	92
PCIe エンドポイントデバイス構成の計画	92
ルートドメインのリポート	93
PCIe ハードウェアの変更	94
▼ 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	95
PCIe SR-IOV 仮想機能の使用	100
SR-IOV の概要	100
SR-IOV のハードウェア要件とソフトウェア要件	102
SR-IOV 機能の現在の制限事項	103
PCIe SR-IOV 仮想機能の使用の計画	103
仮想機能の作成、変更、および破棄	105
I/O ドメイン上の仮想機能の追加および削除	109
SR-IOV: ルートドメインのリポート	111
I/O ドメインを作成するための SR-IOV 仮想機能の使用	111

SR-IOVの詳細なトピック	114
7 仮想ディスクの使用	119
仮想ディスクの概要	119
仮想ディスクの管理	120
▼ 仮想ディスクを追加する方法	121
▼ 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	121
▼ 仮想ディスクオプションを変更する方法	122
▼ タイムアウトオプションを変更する方法	122
▼ 仮想ディスクを削除する方法	122
仮想ディスクの識別子とデバイス名	123
仮想ディスクの表示	124
フルディスク	124
1つのスライスディスク	124
仮想ディスクバックエンドオプション	125
読み取り専用 (ro) オプション	125
排他 (excl) オプション	125
スライス (slice) オプション	126
仮想ディスクバックエンド	127
物理ディスクまたはディスクの LUN	127
▼ 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	127
物理ディスクスライス	128
▼ 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	128
▼ スライス2をエクスポートする方法	129
ファイルおよびボリューム	129
仮想ディスクマルチパスの構成	134
仮想ディスクマルチパスおよび仮想ディスクのタイムアウト	135
▼ 仮想ディスクマルチパスを構成する方法	136
CD、DVD および ISO イメージ	137
▼ 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	138
▼ 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	139
仮想ディスクのタイムアウト	141
仮想ディスクおよび SCSI	141
仮想ディスクおよび format コマンド	142

仮想ディスクと ZFS の使用	142
サービスドメインでの ZFS プールの構成	143
ZFS を使用したディスクイメージの格納	143
ディスクイメージのスナップショットの作成	144
複製を使用して新規ドメインをプロビジョニングする	145
Logical Domains 環境でのボリュームマネージャーの使用	147
ボリュームマネージャーでの仮想ディスクの使用	147
仮想ディスクでのボリュームマネージャーの使用	149
8 仮想ネットワークの使用	151
仮想ネットワークの概要	152
Oracle Solaris 10 ネットワークの概要	152
Oracle Solaris 11 ネットワークの概要	154
仮想スイッチ	157
仮想ネットワークデバイス	158
Inter-Vnet LDC チャネル	158
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	160
▼ Oracle Solaris OS ネットワークインタフェース名を検索する方法	162
自動または手動による MAC アドレスの割り当て	163
Logical Domains に割り当てられる MAC アドレスの範囲	163
自動割り当てのアルゴリズム	164
重複した MAC アドレスの検出	164
解放された MAC アドレス	165
Logical Domains でのネットワークアダプタの使用	166
▼ ネットワークアダプタが GLDv3 準拠かどうかを判別する方法 (Oracle Solaris 10)	166
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	167
Oracle Solaris 10 システムでの NAT の構成	167
Oracle Solaris 11 システムでの NAT の構成	169
Logical Domains 環境での IPMP の構成	171
ドメインの IPMP グループへの仮想ネットワークデバイスの構成	171
サービスドメインでの IPMP の構成と使用	172
Logical Domains 仮想ネットワークでのリンクベースの IPMP の使用	173
Logical Domains 1.3 以前のリリースの IPMP の構成および使用	177
VLAN のタグ付けの使用	179

ポート VLAN ID (PVID)	180
VLAN ID (VID)	180
▼ VLAN を仮想スイッチおよび仮想ネットワークデバイスに割り当てる方法	181
▼ 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	182
NIU ハイブリッド I/O の使用	183
▼ 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	186
▼ ハイブリッドモードを有効にする方法	186
▼ ハイブリッドモードを無効にする方法	186
仮想スイッチでのリンクアグリゲーションの使用	187
ジャンボフレームの構成	188
▼ 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	189
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)	192
Oracle Solaris 11 のネットワーク固有の機能の相違点	193
9 ドメインの移行	195
ドメインの移行の概要	196
移行処理の概要	196
ソフトウェアの互換性	197
移行処理のセキュリティー	197
ドメインの移行	198
予行演習の実行	198
対話型でない移行の実行	198
アクティブなドメインの移行	199
CPU のドメイン移行要件	199
メモリーの移行要件	200
物理 I/O デバイスの移行要件	201
仮想 I/O デバイスの移行要件	201
NIU ハイブリッド I/O の移行要件	202
暗号化装置の移行要件	203
アクティブなドメインの遅延再構成	203
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	203
ほかのドメインの操作	204

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	204
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	204
仮想 I/O デバイスの移行要件	205
PCIe エンドポイントデバイスの移行要件	205
進行中の移行のモニタリング	205
進行中の移行の取り消し	206
移行の失敗からの回復	206
移行の例	207
10 リソースの管理	209
リソースの再構成	209
動的再構成	209
遅延再構成	210
リソースの割り当て	211
CPU の割り当て	211
▼ コア全体の制約を適用する方法	211
▼ コアの最大数の制約を適用する方法	212
コア全体の制約と他のドメイン機能の相互作用	213
ハードパーティションによるシステムの構成	215
ドメインの構成のチェック	216
CPU コア全体によるドメインの構成	217
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	220
ドメインへの物理リソースの割り当て	222
▼ physical-bindings 制約を削除する方法	224
▼ 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	225
制御ドメインでの物理リソースの管理	225
ドメインでの物理リソース管理の制限	226
メモリーの動的再構成の使用	226
メモリーの追加	227
メモリーの削除	227
部分的なメモリー DR 要求	228
制御ドメインのメモリーの再構成	228
動的再構成と遅延再構成	229
メモリー配置	229

メモリー DR の例	231
電源管理の使用	234
動的なリソース管理の使用	234
ドメインリソースの一覧表示	238
マシンが読み取り可能な出力	238
フラグの定義	239
利用統計情報の定義	239
さまざまなリストの表示	240
制約の一覧表示	243
11 ドメイン構成の管理	245
将来の再構築用のドメイン構成の保存	245
▼ドメイン構成を保存する方法	246
▼XML ファイルからのドメイン構成の復元方法 (ldm add-domain)	246
▼XML ファイルからのドメイン構成の復元方法 (ldm init-system)	247
Logical Domains 構成の管理	248
▼自動回復ポリシーを変更する方法	250
12 その他の管理タスクの実行	253
CLI での名前を入力	253
ファイル名 (<i>file</i>) と変数名 (<i>var-name</i>)	253
仮想ディスクサーバー <i>backend</i> および仮想スイッチデバイス名	254
構成名 (<i>config-name</i>)	254
その他のすべての名前	254
ネットワークを介したゲストコンソールへの接続	254
コンソールグループの使用	255
▼複数のコンソールを1つのグループにまとめる方法	255
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	256
Oracle VM Server for SPARC による Oracle Solaris OS の運用	257
Oracle Solaris OS の起動後には OpenBoot ファームウェアを使用できない	257
サーバーの電源再投入の実行	257
電源管理されているドメインのアクティブな CPU での <code>psradm(1M)</code> コマンドの使用禁止	258
Oracle Solaris OS ブレークの結果	258
制御ドメインの停止またはリブートの結果	258

サービスプロセッサでの Logical Domains の使用	259
▼ドメインの構成をデフォルトまたは別の構成にリセットする方法	259
ドメインの依存関係の構成	260
ドメインの依存関係の例	261
依存サイクル	262
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	264
CPU マッピング	264
メモリアドレスのマッピング	264
CPU およびメモリアドレスのマッピングの例	265
ユニバーサル固有識別子の使用	266
仮想ドメイン情報コマンドと API	267
パート II オプションの Oracle VM Server for SPARC ソフトウェア	269
13 Oracle VM Server for SPARC 物理から仮想への変換ツール	271
Oracle VM Server for SPARC P2V ツールの概要	271
収集フェーズ	272
準備フェーズ	273
変換フェーズ	273
バックエンドデバイス	274
Oracle VM Server for SPARC P2V ツールのインストール	275
準備すべき事柄	275
制限事項	276
▼Oracle VM Server for SPARC P2V ツールのインストール方法	276
ldmp2v コマンドの使用	278
14 Oracle VM Server for SPARC Configuration Assistant (Oracle Solaris 10)	285
Configuration Assistant (ldmconfig) の使用	285
Configuration Assistant のインストール	285
ldmconfig の機能	286
15 電源管理の使用	291
電源管理の使用	291

16 Oracle VM Server for SPARC 管理情報ベース (MIB) ソフトウェアの使用	295
Oracle VM Server for SPARC 管理情報ベースの概要	296
ソフトウェアコンポーネント	296
システム管理エージェント	297
Logical Domains Manager と Oracle VM Server for SPARC MIB	298
Oracle VM Server for SPARC MIB オブジェクトツリー	298
Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成	299
Oracle VM Server for SPARC MIB ソフトウェアのインストールと構成 (タスク マップ)	300
セキュリティの管理	303
▼最初の snmpv3 ユーザーを作成する方法	303
ドメインのモニタリング	305
環境変数の設定	305
Oracle VM Server for SPARC MIB の問い合わせ	306
Oracle VM Server for SPARC MIB 情報の取得	308
SNMP トラップの使用	328
Oracle VM Server for SPARC MIB モジュールトラップの使用	329
Oracle VM Server for SPARC MIB トラップの説明	331
ドメインの起動と停止	336
ドメインの起動と停止	337
17 Logical Domains Manager の検出	341
Logical Domains Manager を実行しているシステムの検出	341
マルチキャスト通信	341
メッセージ形式	342
▼サブネット上で実行している Logical Domains Manager を検出する方法	342
18 Logical Domains Manager での XML インタフェースの使用	345
XML トランスポート	345
XMPP サーバー	346
ローカル接続	346
XML プロトコル	346
要求メッセージと応答メッセージ	347
イベントメッセージ	351
登録および登録解除	351

<LDM_event> メッセージ	352
イベントタイプ	353
Logical Domains Manager の処理	356
Logical Domains Manager のリソースとプロパティ	358
ドメインの情報 (ldom_info) リソース	358
CPU (cpu) リソース	359
MAU (mau) リソース	362
メモリー (memory) リソース	362
仮想ディスクサーバー (vds) リソース	363
仮想ディスクサーバーボリューム (vds_volume) リソース	363
ディスク (disk) リソース	364
仮想スイッチ (vsw) リソース	364
ネットワーク (network) リソース	366
仮想コンソール端末集配信装置 (vcc) リソース	367
変数 (var) リソース	367
物理 I/O デバイス (physio_device) リソース	368
SP 構成 (spconfig) リソース	368
DRM ポリシー構成 (policy) リソース	369
仮想データプレーンチャンネルサービス (vdpcs) リソース	370
仮想データプレーンチャンネルクライアント (vdpcc) リソース	370
コンソール (console) リソース	371
ドメインの移行	372
XML スキーマ	373
用語集	375
索引	385

はじめに

Oracle VM Server for SPARC 3.0 管理ガイドでは、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 3.0 ソフトウェアの概要、セキュリティー上の考慮事項、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 3.0 リリースノート』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。

このガイドは、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) の実践的な知識がある、これらのサーバーのシステム管理者を対象としています。

注- このマニュアルで説明する機能は、『Oracle VM Server for SPARC 3.0 リリースノート』で一覧表示したサポートされているすべての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で使用可能です。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 3.0 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。

関連ドキュメント

次の表に、Oracle VM Server for SPARC 3.0 リリースで利用できるドキュメントを示します。これらのドキュメントは、特に記載がない限り、HTML 形式と PDF 形式で利用できます。

表 P-1 関連ドキュメント

用途	タイトル
Oracle VM Server for SPARC 3.0 ソフトウェア	<ul style="list-style-type: none"> 『Oracle VM Server for SPARC 3.0 管理ガイド』 『Oracle VM Server for SPARC 3.0 セキュリティーガイド』 『Oracle VM Server for SPARC 3.0 リファレンスマニュアル』 『Oracle VM Server for SPARC 3.0 リリースノート』
Oracle VM Server for SPARC 3.0 drd(1M) および vntsd(1M) マニュアルページ	Oracle Solaris OS リファレンスマニュアル: <ul style="list-style-type: none"> ■ Oracle Solaris 10 Documentation ■ Oracle Solaris 11.1 ドキュメント
Oracle Solaris OS: インストールと構成	Oracle Solaris OS インストールおよび構成ガイド: <ul style="list-style-type: none"> ■ Oracle Solaris 10 Documentation ■ Oracle Solaris 11.1 ドキュメント
Oracle VM Server for SPARC および Oracle Solaris OS のセキュリティー	Oracle VM Server for SPARC のホワイトペーパーおよび Oracle Solaris OS セキュリティーガイド: <ul style="list-style-type: none"> ■ Secure Deployment of Oracle VM Server for SPARC (http://www.oracle.com/technetwork/articles/systems-hardware-architecture/secure-ovm-sparc-deployment-294062.pdf) ■ 『Oracle Solaris 10 Security Guidelines』 ■ 『Oracle Solaris 11 Security Guidelines』

使用しているサーバー、ソフトウェア、または Oracle Solaris OS に関連するドキュメントは、<http://www.oracle.com/technetwork/indexes/documentation/index.html> で参照できます。必要なドキュメントや情報を検索するには、「Search」ボックスを使用します。

Oracle サポートへのアクセス

Oracle ユーザーは My Oracle Support から電子サポートにアクセスできます。詳細については、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> にアクセスしてください。または、聴覚に障害がある場合は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> にアクセスしてください。

表記上の規則

次の表では、このマニュアルで使用される表記上の規則について説明します。

表 P-2 表記上の規則

字体	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力を示します。	.login ファイルを編集します。 ls-a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	可変部分: 実際に使用する特定の名前または値で置き換えます。	ファイルを削除するコマンドは、rm <i>filename</i> です。
AaBbCc123	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第6章を参照してください。 キャッシュは、ローカルに格納されたコピーです。 ファイルを保存しないでください。 注: 一部の強調された項目はオンラインでは太字で表示されます。

コマンド例のシェルプロンプト

次の表に、Oracle Solaris OS に含まれるシェルのデフォルトの UNIX システムプロンプトおよびスーパーユーザーのプロンプトを示します。コマンド例に示されるデフォルトのシステムプロンプトは Oracle Solaris のリリースに応じて異なることに注意してください。

表 P-3 シェルプロンプトについて

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
スーパーユーザーの Bash シェル、Korn シェル、および Bourne シェル	#

表 P-3 シェルプロンプトについて (続き)

シェル	プロンプト
Cシェル	machine_name%
スーパーユーザーのCシェル	machine_name#

パート I

Oracle VM Server for SPARC 3.0 ソフトウェア

ここでは、SPARC T シリーズおよび Fujitsu M10 サーバー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 3.0 ソフトウェアを紹介します。

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 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。詳細については、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』の「[必須および推奨される Oracle Solaris OS バージョン](#)」を参照してください。

Oracle VM Server for SPARC は、SPARC T シリーズおよび Fujitsu M10 システム 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 Fujitsu M10 システム 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。

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 10 OS を実行している場合でも、ゲストドメインで Oracle Solaris 11 OS を実行することができます。同様に、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 Oracle Solaris 10 OS を実行することができます。

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。

この章では、次の項目について説明します。

- 22 ページの「[ハイパーバイザと Logical Domains](#)」
- 24 ページの「[Logical Domains Manager](#)」
- 28 ページの「[Oracle VM Server for SPARC 物理から仮想への変換ツール](#)」
- 28 ページの「[Oracle VM Server for SPARC Configuration Assistant](#)」
- 29 ページの「[Oracle VM Server for SPARC 管理情報ベース](#)」

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。

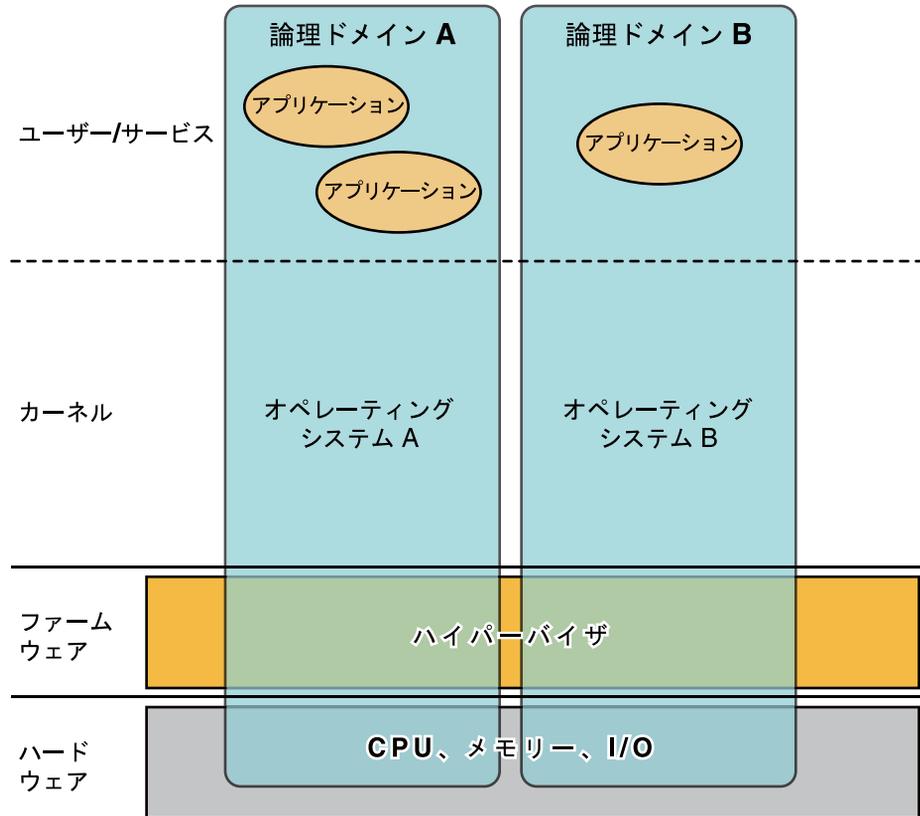
SPARC ハイパーバイザは、小さなファームウェア層で、オペレーティングシステムを記述できる安定した仮想化マシンアーキテクチャーを提供します。ハイパーバイザを使用する SPARC サーバーでは、論理オペレーティングシステムの動作をハイパーバイザが制御できるようにするためのハードウェア機能が用意されています。

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、Logical Domains の機能を構成する次の層についても示します。

- ユーザー/サービスまたはアプリケーション
- カーネルまたはオペレーティングシステム
- ファームウェアまたはハイパーバイザ
- 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。これにより、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ーフェースです。このユーザーインター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。Oracle VM Manager 3.2 および Oracle VM Server for SPARC 3.0 以降では、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せん。
- 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他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 PCIe ルートコンプレックスを所有するか、直接 I/O (DIO) 機能を使用して PCIe スロットまたはオンボードの PCIe デバイスを所有することができます。[89 ページの「PCIe エンドポイントデバイスの割り当て」](#)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、Oracle Sun SPARC Enterprise T5440 サーバーを使用している場合、最大で4つのルートドメインを保持できます。

- 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。

既に Logical Domains 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、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。これを行うには、ドメインを追加し、制御ドメインから新しいドメインにアプリケーションを移動します。

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\(1M\)](#) 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 デーモン `ldmd` が実行されている必要があります。

仮想入出力

Logical Domains 環境では、システムで最大 128 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 スロットの数の制限があります。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。[第 6 章「I/O ドメインの設定」](#) を参照してください。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。

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

Logical Domains 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

Logical Domains 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、制御ドメインのリポ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、[209 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。[ldm\(1M\)](#) マニュアルページを参照してください。SP からブートする構成を指定することもできます。[259 ページの「サービスプロセッサでの Logical Domains の使用」](#)を参照してください。

構成の管理については、248 ページの「[Logical Domains 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 10 OS または 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することができます。

- 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 を実行しており、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 - 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 13 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 Configuration Assistant

Oracle VM Server for SPARC Configuration Assistant を使用すると、基本的なプロパティを設定することによって論理ドメインの構成手順を実行できます。Oracle VM Server for SPARC ソフトウェアがインストールされているが、構成されていない任意のシステムを構成するために使用できます。

Configuration Assistant は、構成データを収集したあと、論理ドメインとしてブートするのに適した構成を作成します。Configuration Assistant によって選択されるデフォルト値を使用して、有効なシステム構成を作成することもできます。

注 - `ldmconfig` コマンドは Oracle Solaris 10 システムでのみサポートされます。

Configuration Assistant は端末ベースのツールです。

詳細については、第 14 章「[Oracle VM Server for SPARC Configuration Assistant \(Oracle Solaris 10\)](#)」および [ldmconfig\(1M\)](#) 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 (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 16 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。

ソフトウェアのインストールおよび有効化

この章では、Oracle VM Server for SPARC 3.0 ソフトウェアを有効にするために必要なさまざまなソフトウェアコンポーネントをインストールまたはアップグレード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。

- 31 ページの「[必須の Oracle VM Server for SPARC ソフトウェアコンポーネント](#)」
- 32 ページの「[新しいシステムへの Oracle VM Server for SPARC ソフトウェアのインストール](#)」
- 38 ページの「[Oracle VM Server for SPARC をすでに使用しているシステムのアップグレード](#)」
- 44 ページの「[出荷時デフォルト構成と Logical Domains の無効化](#)」

必須の Oracle VM Server for SPARC ソフトウェアコンポーネント

Oracle VM Server for SPARC ソフトウェアを使用するには、次のコンポーネントが必要です。

- サポートされるプラットフォーム。サポートされるプラットフォームの一覧については、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』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。サポートされているファームウェアについては、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』の「[Oracle VM Server for SPARC 3.0 の機能を有効にするための必須ソフトウェア](#)」および 33 ページの「[システムファームウェアのアップグレード](#)」を参照してください。
- 『[Oracle VM Server for SPARC 3.0 リリースノート](#)』の「[必須のソフトウェアとパッチ](#)」で推奨されているすべてのパッチが適用された、Oracle Solaris 11 OS 以上と同等のオペレーティングシステムおよび Support Repository Update (SRU)、または、該当する場合は Oracle Solaris 10 8/11 OS が動作している制御ドメイン。39 ページの「[Oracle Solaris OS のアップグレード](#)」を参照してください。

- 制御ドメインにインストールされて有効になっている Oracle VM Server for SPARC 3.0 ソフトウェア。35 ページの「[Logical Domains Manager のインストール](#)」を参照してください。
- (オプション) Oracle VM Server for SPARC 管理情報ベース (MIB) ソフトウェア パッケージ。第 16 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。

Logical Domains Manager をインストールまたはアップグレードする前に、Oracle Solaris OS およびシステムファームウェアが、使用しているサーバーでインストールまたはアップグレードされている必要があります。システムがすでに Oracle VM Server for SPARC ソフトウェアを使用している場合は、38 ページの「[Oracle VM Server for SPARC をすでに使用しているシステムのアップグレード](#)」を参照してください。そうでない場合は、32 ページの「[新しいシステムへの Oracle VM Server for SPARC ソフトウェアのインストール](#)」を参照してください。

新しいシステムへの Oracle VM Server for SPARC ソフトウェアのインストール

Oracle VM Server for SPARC ソフトウェアをサポートする SPARC プラットフォームは、Oracle Solaris 10 OS または Oracle Solaris 11 OS がプリインストールされた状態で出荷されます。初期状態では、プラットフォームは 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示されます。Oracle Solaris OS、システムファームウェア、および Logical Domains Manager をインストールすると、Oracle Solaris OS の元のシステムおよびインスタンスが制御ドメインになります。プラットフォームのこの最初のドメインには、primary という名前が付けられます。この名前を変更したり、このドメインを削除したりすることはできません。このドメインから、Oracle Solaris OS のさまざまなインスタンスをホストする複数のドメインを持つようにプラットフォームを再構成できます。

注- ゲストドメインで動作するバージョンの Oracle Solaris OS ソフトウェアは、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せん。そのため、primary ドメインで Oracle Solaris 10 OS を実行している場合でも、ゲストドメインのいずれかで Oracle Solaris 11 OS を実行できます。同様に、primary ドメインで Oracle Solaris 11 OS を実行している場合でも、ゲストドメインのいずれかで Oracle Solaris 10 OS を実行できます。

primary ドメインでどの Oracle Solaris OS バージョンを実行するかは、要件と、Oracle Solaris 10 と Oracle Solaris 11 の間の潜在的な機能の違いに基づいて決定してください。『[Oracle Solaris 11.1 Release Notes](#)』および『[Transitioning From Oracle Solaris 10 JumpStart to Oracle Solaris 11.1 Automated Installer](#)』を参照してください。

Oracle Solaris OS の更新

新しいシステムで独自のインストール方針に準拠させるには、出荷時にインストールされた OS の再インストールが必要な場合があります。『Oracle VM Server for SPARC 3.0 リリースノート』の「必須および推奨される Oracle Solaris OS バージョン」を参照してください。Oracle Solaris OS の包括的なインストール手順については、Oracle Solaris 10 8/11 Information Library (http://docs.oracle.com/cd/E23823_01/) および Oracle Solaris 11.1 Information Library (http://docs.oracle.com/cd/E23824_01/) を参照してください。インストール内容は、使用するシステムの要件に合わせて調整できます。

ご利用のシステムに Oracle Solaris OS がインストールされている場合は、Oracle VM Server for SPARC 3.0 ソフトウェアに関連付けられた OS のバージョンにアップグレードする必要があります。『Oracle VM Server for SPARC 3.0 リリースノート』の「必須のソフトウェアとパッチ」を参照してください。Oracle Solaris OS の包括的なアップグレード手順については、Oracle Solaris 10 8/11 Information Library (http://docs.oracle.com/cd/E23823_01/) および Oracle Solaris 11.1 Information Library (http://docs.oracle.com/cd/E23824_01/) を参照してください。

システムファームウェアのアップグレード

次のタスクでは、Integrated Lights Out Manager (ILOM) ソフトウェアを使用して、SPARC T シリーズシステム上のシステムファームウェアをアップグレードする方法を示します。次のリソースも参照してください。

- ILOM ソフトウェアを使用したシステムファームウェアのアップグレードについては、『Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI 手順ガイド』の「ファームウェアの更新」および「ILOM ファームウェアの更新」を参照してください。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。
- 使用しているプラットフォームのシステムファームウェアは、<http://www.oracle.com/technetwork/systems/patches/firmware/index.html> から入手できます。
- サポートされるサーバーに必要なシステムファームウェアについては、『Oracle VM Server for SPARC 3.0 リリースノート』の「必須および推奨されるシステムファームウェアパッチ」を参照してください。
- 制御ドメインからシステムファームウェアをアップグレードする場合は、SPARC T-Series Servers Documentation (<http://www.oracle.com/technetwork/documentation/sparc-tseries-servers-252697.html>) で入手可能なシステムファームウェアのプロダクトノートを参照してください。

- サポートされるサーバーのシステムファームウェアのインストールおよびアップグレードについては、そのサーバーの管理ガイドまたはプロダクトノートを参照してください。
- ILOM Web インタフェースを使用してシステムファームウェアをアップグレードすることもできます。『Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface 手順ガイド』の「ILOM ファームウェアの更新」を参照してください。

eXtended System Control Facility (XSCF) を使用して、Fujitsu M10 システム ファームウェアをアップグレードするには、次のリソースを参照してください。

- Fujitsu M10 システム システム運用・管理ガイド
- Fujitsu M10 システム XSCF リファレンスマニュアル

Logical Domains Manager のダウンロード

Oracle Solaris 10 OS および Oracle Solaris 11 OS の両方の最新パッケージを入手できます。Oracle VM Server for SPARC ソフトウェアは、デフォルトでは Oracle Solaris 11 OS に同梱されていることに注意してください。

- **Oracle Solaris 10 OS。** 「My Oracle Support」から `OVM_Server_SPARC-3_0.zip` パッケージをダウンロードします。34 ページの「[Logical Domains Manager ソフトウェアをダウンロードする方法 \(Oracle Solaris 10\)](#)」を参照してください。
- **Oracle Solaris 11 OS。** Oracle Solaris 11 Support Repository から `ldomsmanager` パッケージを入手します。42 ページの「[Oracle VM Server for SPARC 3.0 ソフトウェアにアップグレードする方法 \(Oracle Solaris 11\)](#)」を参照してください。

▼ Logical Domains Manager ソフトウェアをダウンロードする方法 (Oracle Solaris 10)

- 1 **zip** ファイル (`OVM_Server_SPARC-3_0.zip`) をダウンロードします。
ソフトウェアは、<http://www.oracle.com/virtualization/index.html> で入手できます。
- 2 保存した **zip** ファイルを解凍します。

```
$ unzip OVM_Server_SPARC-3_0.zip
```

ファイルの構造およびファイルの内容については、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』の「[Oracle VM Server for SPARC 3.0 ソフトウェアの場所](#)」を参照してください。

Logical Domains Manager のインストール

次に、Logical Domains Manager ソフトウェアのインストール方法を示します。

- **Oracle Solaris 10** のみ。インストールスクリプトを使用してパッケージおよびパッチをインストールします。この方法では Logical Domains Manager ソフトウェアが自動的にインストールされます。35 ページの「(Oracle Solaris 10) Logical Domains Manager ソフトウェアの自動的なインストール」を参照してください。
- **Oracle Solaris 10** のみ。Oracle Solaris JumpStart 機能を使用して、パッケージをネットワークインストールの一部としてインストールします。JumpStart サーバーの構成の詳細については、『Oracle Solaris 10 8/11 Installation Guide: Custom JumpStart and Advanced Installations』を参照してください。この機能の使用法の詳細については、『JumpStart Technology: Effective Use in the Solaris Operating Environment』も参照してください。
- **Oracle Solaris 11** のみ。Oracle Solaris 11 Automated Installer 機能を使用して、パッケージをネットワークインストールの一部としてインストールします。『Installing Oracle Solaris 11.1 Systems』の「How to Use the Automated Installer」および『Transitioning From Oracle Solaris 10 JumpStart to Oracle Solaris 11.1 Automated Installer』を参照してください。
- パッケージを手動でインストールします。36 ページの「Logical Domains Manager ソフトウェアの手動によるインストール」を参照してください。

注 - Oracle VM Server for SPARC パッケージをインストールしたあとで、Oracle VM Server for SPARC MIB ソフトウェアパッケージを手動でインストールする必要があります。これは、ほかのパッケージとともに自動的にインストールされません。Oracle VM Server for SPARC MIB のインストールおよび使用については、第 16 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 Solaris 10) Logical Domains Manager ソフトウェアの自動的なインストール

install-ldm インストールスクリプトを使用する場合、スクリプトの実行方法を指定する選択肢がいくつかあります。それぞれの選択肢について、次の手順で説明します。

- オプションを指定せずに install-ldm スクリプトを使用すると、自動的に次の処理を行います。
 - Oracle Solaris OS リリースが Oracle Solaris 10 OS であることを確認します
 - パッケージのサブディレクトリである SUNWldm/ および SUNWldmp2v/ が存在することを確認します。
 - 前提条件となる Logical Domains ドライバパッケージの SUNWldomr および SUNWldomu が存在することを確認します。

- SUNWldm および SUNWldmp2v パッケージがインストールされていないことを確認します。
- Oracle VM Server for SPARC 3.0 ソフトウェアをインストールします。
- すべてのパッケージがインストールされていることを確認します。
- SST (SUNWjass) がすでにインストールされている場合は、制御ドメインの Oracle Solaris OS の強化を求めるプロンプトが表示されます。
- Oracle VM Server for SPARC Configuration Assistant (ldmconfig) を使用してインストールを実行するかどうかを判断します。
- -c オプションを指定して install-ldm スクリプトを使用すると、ソフトウェアのインストール後に自動的に **Oracle VM Server for SPARC Configuration Assistant** が実行されます。
- -s オプションを指定して install-ldm スクリプトを使用すると、**Oracle VM Server for SPARC Configuration Assistant** の実行はスキップされます。
- SST ソフトウェアとともに install-ldm スクリプトおよび次のオプションを使用すると、次の操作を実行できます。
 - install-ldm -d. -secure.driver で終わるドライバ以外の SST ドライバを指定できます。このオプションは、前述の選択肢で示したすべての機能を自動的に実行し、指定した SST のカスタマイズドライバ(たとえば server-secure-myname.driver) で制御ドメインの Oracle Solaris OS を強化します。
 - install-ldm -d none。SST を使用して制御ドメインで実行している Oracle Solaris OS を強化しないことを指定します。このオプションは、前述の選択肢で示した強化以外のすべての機能を自動的に実行します。SST の使用を省略することはお勧めしません。別の処理を使用して制御ドメインを強化する場合にすぎ、この使用を省略するようにしてください。
 - install-ldm -p。インストール後の処理 (Logical Domains Manager デーモン (ldmd) の有効化および SST の実行) のみを実行することを指定します。たとえば、SUNWldm および SUNWjass パッケージがサーバーにプリインストールされている場合に、このオプションを使用します。

Logical Domains Manager ソフトウェアの手動によるインストール

次に、Oracle Solaris 10 OS 上に Oracle VM Server for SPARC 3.0 ソフトウェアを手動でインストールする手順を説明します。

Oracle Solaris 11 OS をインストールすると、デフォルトで Oracle VM Server for SPARC 2.1 ソフトウェアがインストールされます。Oracle VM Server for SPARC 3.0 ソフトウェアをインストールする場合は、[42 ページの「Oracle VM Server for SPARC 3.0 ソフトウェアにアップグレードする方法 \(Oracle Solaris 11\)」](#)を参照してください。

▼ Oracle VM Server for SPARC 3.0 ソフトウェアを手動でインストールする方法 (Oracle Solaris 10)

始める前に Oracle VM Server for SPARC 3.0 ソフトウェア (SUNWldm および SUNWldmp2v パッケージ) をダウンロードします。詳細は、34 ページの「[Logical Domains Manager ソフトウェアをダウンロードする方法 \(Oracle Solaris 10\)](#)」を参照してください。

- 1 (オプション) 必要に応じて、構成をサービスプロセッサ (SP) に保存します。
この手順は、旧バージョンの Oracle VM Server for SPARC ソフトウェアをすでに実行している場合のみ実行してください。

```
primary# ldm add-config config-name
```

- 2 SUNWldm.v および SUNWldmp2v パッケージをインストールします。

```
# pkgadd -Gd . SUNWldm.v SUNWldmp2v
```

対話型プロンプトのすべての質問に対して、y (はい) と答えます。

-G オプションはパッケージを大域ゾーンのみインストールします。-d オプションは、SUNWldm.v および SUNWldmp2v パッケージが含まれるディレクトリのパスを指定します。

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

- 3 SUNWldm および SUNWldmp2v パッケージがインストールされていることを確認します。
リビジョン (REV) 情報の例を次に示します。

```
# pkginfo -l SUNWldm | grep VERSION
VERSION=3.0,REV=2012.11.01.10.20
```

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

Logical Domains Manager デーモンの有効化

install-ldm インストールスクリプトを使用すると、Logical Domains Manager デーモン (ldmd) が自動的に有効になります。Oracle VM Server for SPARC ソフトウェアパッケージをインストールした場合も、ldmd デーモンは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。

▼ 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 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。

```
# svcadm enable ldmd
```

- 2 **ldm list** コマンドを使用して、**Logical Domains Manager** デーモン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が、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。

```
# /opt/SUNWldm/bin/ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active ---c-  SP    64    3264M   0.3%  19d 9m
```

Oracle VM Server for SPARC をすでに使用しているシステムのアップグレード

このセクションでは、Oracle VM Server for SPARC ソフトウェアをすでに使用しているシステムで Oracle Solaris OS、ファームウェア、および Logical Domains Manager コンポーネントをアップグレードするプロセスについて説明します。

使用しているシステムがすでに Oracle VM Server for SPARC ソフトウェアで構成されている場合は、制御ドメインと既存のドメインをアップグレードします。そのようなアップグレードは、それらのドメインですべての Oracle VM Server for SPARC 3.0 機能を使用可能にします。

注 - Oracle VM Server for SPARC ソフトウェアをアップグレードする前に、次の手順を実行してください。

- 必要なシステムファームウェアでシステムをアップグレードします。
『Oracle VM Server for SPARC 3.0 リリースノート』の「Oracle VM Server for SPARC 3.0 の機能を有効にするための必須ソフトウェア」を参照してください。
 - Oracle Solaris 10 OS の必須パッチまたは Oracle Solaris 11 OS SRU を適用します。
『Oracle VM Server for SPARC 3.0 リリースノート』の「必須および推奨される Oracle Solaris OS バージョン」を参照してください。
 - 構成を SP に保存します。
-

Oracle Solaris OS のアップグレード

このバージョンの Oracle VM Server for SPARC ソフトウェア用に使用する必要のある Oracle Solaris 10 または Oracle Solaris 11 OS、および各種ドメインの必須パッチと推奨されるパッチを入手するには、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』の「[必須のソフトウェアとパッチ](#)」を参照してください。Oracle Solaris OS のアップグレードの詳細な手順については、Oracle Solaris 10 および Oracle Solaris 11 のインストールガイドを参照してください。

制御ドメインで Oracle Solaris OS を再インストールする場合は、このセクションに示すとおり、Logical Domains の自動保存構成データおよび制約データベースファイルを保存し復元する必要があります。

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。制御ドメインでオペレーティングシステムを再インストールするたびに、Logical Domains の自動保存構成データを保存し復元する必要があります。このデータは、`/var/opt/SUNWldm/autosave-autosave-name` ディレクトリに格納されています。

`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-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は、[248 ページ](#)の「[Logical Domains 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

この手順は、自動保存ディレクトリを保存および復元する方法を示します。

- 1 自動保存ディレクトリを保存します。

```
# cd /  
# tar -cvpf autosave.tar var/opt/SUNWldm/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/opt/SUNWldm/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/opt/SUNWldm 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

制御ドメインでオペレーティングシステムをアップグレードするたびに、/var/opt/SUNWldm/ldom-db.xml で参照できる Logical Domains の制約データベースファイルを保存し復元する必要があります。

注-また、ディスクスワップなど、制御ドメインのファイルデータを破損するその他の操作を行うときは、/var/opt/SUNWldm/ldom-db.xml ファイルも保存および復元します。

Oracle Solaris 10 Live Upgrade 機能を使用する場合の Logical Domains の制約データベースファイルの保持

制御ドメインで Oracle Solaris 10 Live Upgrade 機能を使用する場合は、/etc/lu/synclist ファイルに次の行を追加することを検討してください。

```
/var/opt/SUNWldm/ldom-db.xml OVERWRITE
```

この行によって、ブート環境を切り替えたときに、データベースがアクティブなブート環境から新しいブート環境に自動的にコピーされます。/etc/lu/synclist と、ブート環境間でのファイルの同期の詳細は、『[Oracle Solaris 10 8/11 Installation Guide: Live Upgrade and Upgrade Planning](#)』の「[Synchronizing Files Between Boot Environments](#)」を参照してください。

Logical Domains Manager およびシステムファームウェアのアップグレード

このセクションでは、Oracle VM Server for SPARC 3.0 ソフトウェアにアップグレードする方法について説明します。

最初に Logical Domains Manager を制御ドメインにダウンロードします。34 ページの「[Logical Domains Manager のダウンロード](#)」を参照してください。

次に、プラットフォーム上で動作している制御ドメイン以外のすべてのドメインを停止します。

▼ プラットフォーム上で動作している制御ドメイン以外のすべてのドメインを停止する方法

このタスクは、システムの電源の再投入またはファームウェアのアップデートを行う予定がある場合にのみ実行してください。Logical Domains Manager ソフトウェアのアップデートのみを行う場合、このタスクを実行する必要はありません。

- 1 各ドメインで **ok** プロンプトに移行します。
- 2 **--a** オプションを使用して、すべてのドメインを停止します。

```
primary# ldm stop-domain -a
```
- 3 制御ドメインから各ドメインに対して **unbind-domain** サブコマンドを実行します。

```
primary# ldm unbind-domain ldom
```

Oracle VM Server for SPARC 3.0 ソフトウェアへのアップグレード

このセクションでは、Oracle VM Server for SPARC 3.0 ソフトウェアにアップグレードする方法について説明します。

▼ Oracle VM Server for SPARC 3.0 ソフトウェアにアップグレードする方法 (Oracle Solaris 10)

- 1 システムファームウェアをフラッシュアップグレードします。
- 2 Logical Domains Manager デーモン (**ldmd**) を無効にします。

```
# svcadm disable ldmd
```
- 3 古い **SUNWldm** パッケージを削除します。

```
# pkgrm SUNWldm
```
- 4 新しい **SUNWldm** パッケージを追加します。

```
# pkgadd -Gd . SUNWldm.v
```

-d オプションの指定は、パッケージが現在のディレクトリに存在することを示します。

- 5 **ldm list** コマンドを使用して、**Logical Domains Manager** デーモン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が、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    32    3264M   0.3%  19d 9m
```

▼ Oracle VM Server for SPARC 3.0 ソフトウェアにアップグレードする方法 (Oracle Solaris 11)

- 1 **Logical Domains Manager** のアップグレード用のドメインを準備します。

次の手順を実行すると、必要に応じて、以前のリリースの Oracle VM Server for SPARC ソフトウェアを実行するブート環境 (BE) に「ロールバック」することができます。

- a. 構成を **SP** に保存します。

```
# ldm add-config config-name
```

次の例では、**ldoms-prev-config** という構成を保存します。

```
# ldm add-config ldoms-prev-config
```

- b. 既存の **BE** のスナップショットを作成します。

```
# beadm create snapshot-name
```

次の例では、**S10811@ldoms-prev-backup** というスナップショットを作成します。

```
# beadm create S10811@ldoms-prev-backup
```

- c. スナップショットに基づいてバックアップ **BE** を作成します。

```
# beadm create -e snapshot-name BE-name
```

次の例では、**S10811@ldoms-prev-backup** というスナップショットから、**ldoms-prev-backup** という新しい **BE** を作成します。

```
# beadm create -e S10811@ldoms-prev-backup ldoms-prev-backup
```

- 2 オンラインソフトウェアリポジトリを使用するために登録します。

[Certificate Generator Online Help \(https://pkg-register.oracle.com/help/#support\)](https://pkg-register.oracle.com/help/#support) を参照してください。

- 3 最新の SRU への更新の一部として、オンラインソフトウェアリポジトリから、**Oracle VM Server for SPARC 3.0** バージョンの **ldomsmanager** パッケージをインストールします。

```
# pkg update
    Packages to install: 1
    Packages to update: 89
    Create boot environment: No
    Create backup boot environment: No
    Services to change: 3

PHASE                                ACTIONS
Removal Phase                        517/517
Install Phase                        806/806
Update Phase                          5325/5325

PHASE                                ITEMS
Package State Update Phase          179/179
Package Cache Update Phase           89/89
Image State Update Phase              2/2
```

- 4 パッケージがインストールされていることを確認します。

```
# pkg info ldomsmanager
Name: system/ldoms/ldomsmanager
Summary: Logical Domains Manager
Description: LDoms Manager - Virtualization for SPARC T-Series
Category: System/Virtualization
State: Installed
Publisher: solaris
Version: 3.0.0.0
Build Release: 5.11
Branch: 0.175.0.0.0.1.0
Packaging Date: Thu Oct 25 23:06:35 2012
Size: 2.34 MB
FMRI: pkg:///solaris/system/ldoms/ldomsmanager@
3.0.0.0,5.11-0.175.0.0.0.1.0:20120221T141945Z
```

- 5 **ldmd** サービスを再起動します。

```
# svcadm restart ldmd
```

- 6 正しい **ldm** バージョンが実行されていることを確認します。

```
# ldm -V
```

- 7 構成を **SP** に保存します。

```
# ldm add-config config-name
```

次の例では、**ldoms-3.0-config** という構成を保存します。

```
# ldm add-config ldoms-3.0-config
```

出荷時デフォルト構成と Logical Domains の無効化

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る初期構成は、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、他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、Logical Domains 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 `--a` オプションを使用して、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-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
```

▼ すべての論理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論理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 -i1 -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読み込みます。

```
-> stop /SYS  
-> start /SYS
```

▼ Logical Domains Manager を無効にする方法

- 制御ドメインから Logical Domains Manager を無効にします。

```
primary# svcadm disable ldmd
```

注 - 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、電源管理など、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、電源管理の場合にはこの方法は使用できません。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。

▼ Logical Domains Manager を削除する方法

出荷時デフォルト構成を復元して Logical Domains Manager を無効にしたあとで、Logical Domains Manager ソフトウェアを削除できます。

注 - 出荷時デフォルト構成を復元する前に Logical Domains Manager を削除する場合は、次の手順に示すように、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。

- **Logical Domains Manager** ソフトウェアを削除します。
 - **Oracle Solaris 10** の **SUNWldm** および **SUNWldmp2v** パッケージを削除します。
primary# **pkgrm SUNWldm SUNWldmp2v**
 - **Oracle Solaris 11** の **ldomsmanager** パッケージを削除します。
primary# **pkg uninstall ldomsmanager**

- ▼ **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**

出荷時デフォルト構成を復元する前に Logical Domains Manager を削除する場合は、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読み込みます。
-> **reset /SYS**

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。

- 47 ページの「RBAC の使用による論理ドメインの管理の委任」
- 51 ページの「RBAC の使用によるドメインコンソールへのアクセスの制御」
- 59 ページの「監査の有効化と使用」

注- 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。

RBAC 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つの定義済みの役割ベースのアクセス制御 (RBAC) 権利プロファイルを、ローカル RBAC 構成に追加します。これらの権利プロファイルを使用して、特権のないユーザーに次の管理権限を委任できます。

- 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 RBAC 構成に基づいて決定します。『System

Administration Guide: Security Services』または『Oracle Solaris 11.1 Administration: Security Services』のパート III 「Roles, Rights Profiles, and Privileges」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。この章の例はすべて、RBAC 構成がローカルファイルを使用すると仮定しています。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、51 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。

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 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。
- 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。

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 プロファイルに直接割り当てられているユーザーは、プロファイルシェルを起動し、`ldm` コマンドをセキュリティー属性を使用して実行する必要があります。

ます。詳細は、『[System Administration Guide: Security Services](#)』または『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート III 「Roles, Rights Profiles, and Privileges」を参照してください。

1 管理者になります。

Oracle Solaris 10 の場合は、『[System Administration Guide: Security Services](#)』の「[Configuring RBAC \(Task Map\)](#)」を参照してください。Oracle Solaris 11.1 の場合は、『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート III 「Roles, Rights Profiles, and Privileges」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つのセキュリティー階層は、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 10 の場合は、『[System Administration Guide: Security Services](#)』の「[Configuring RBAC \(Task Map\)](#)」を参照してください。Oracle Solaris 11.1 の場合は、『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート III 「Roles, Rights Profiles, and Privileges」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 3-1 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: ldm_read-password
Re-enter new Password: ldm_read-password
passwd: password successfully changed for ldm_read
# useradd -R ldm_read user_1
# passwd user_1
New Password: user_1-password
Re-enter new Password: user_1-password
passwd: password successfully changed for user_1
# su user_1
Password: user_1-password
$ id
uid=95555(user_1) gid=10(staff)
$ roles
ldm_read
$ su ldm_read
Password: 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 RBAC プロファイルをローカルの `/etc/security/prof_attr` ファイルに追加します。

```
LDoms Review:::Review LDoms configuration:profiles=auths=solaris.ldoms.read
LDoms Management:::Manage LDoms domains:profiles=auths=solaris.ldoms.*
```

Logical Domains Manager パッケージは、LDoms Management 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 `/etc/security/exec_attr` ファイルに追加します。

```
LDoms Management:suser:cmd:::/usr/sbin/ldm:privs=file_dac_read,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 3-1 ldm サブコマンドおよびユーザー承認

ldm サブコマンド ¹	ユーザー承認
add-*	solaris.ldoms.write
bind-domain	solaris.ldoms.write
list	solaris.ldoms.read
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

¹ 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。

RBAC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 `auth_attr` 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、53 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54 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、56 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58 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 all_cons
```

- 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 3-2 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 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

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

$ telnet 0 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きるかを示しています。

```

$ id
uid=702048(dana) gid=1(other)
$ telnet 0 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。

- Oracle Solaris 10 OS の場合は、`/etc/security/prof_attr` ファイルを編集します。次のエントリを指定します。

```
LDoms Consoles::Access LDoms Consoles:auths=solaris.vntsd.consoles
```

- Oracle Solaris 11 OS の場合は、`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。

- Oracle Solaris 10 OS の場合は、権利プロファイルをユーザーに割り当てます。

```
primary# usermod -P "All,Basic Solaris User,LDoms Consoles" username
```

LDoms Consoles プロファイルを追加する際は、注意して既存のプロファイルを指定してください。前のコマンドは、ユーザーがすでに All および Basic Solaris User プロファイルを持っていることを示しています。

- Oracle Solaris 11 OS の場合は、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 0 5000
```

例 3-3 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。

- Oracle Solaris 10: /etc/security/prof_attr ファイルに次のエントリを追加して、solaris.vntsd.consoles 承認を持つ権利プロファイルを作成します。

```
LDoms Consoles::Access LDoms Consoles:auths=solaris.vntsd.consoles
```

権利プロファイルを *username* に割り当てます。

```
primary# usermod -P "All,Basic Solaris User,LDoms Consoles" username
```

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 0 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
```

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

- Oracle Solaris 11: profiles コマンドを使用して、/etc/security/prof_attr ファイルに solaris.vntsd.consoles 承認を持つ権利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'
```

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 0 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 1つのドメインに対する承認を `/etc/security/auth_attr` ファイルに追加します。承認名は、ドメインの名前から派生し、`solaris.vntsd.console-domain-name` の形式になります。

```
solaris.vntsd.console-domain-name:::Access domain-name 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-name 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 3-4 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。

最初に、1つのドメイン *ldg1* に対する承認を */etc/security/auth_attr* ファイルに追加します。

```
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 0 5000
Trying 0.0.0.0...
Escape character is '^'].
```

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

次は、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 0 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 1つのドメインに対する承認を `/etc/security/auth_attr` ファイルに追加します。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-name::Access domain-name 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。

- Oracle Solaris 10 OS の場合は、`/etc/security/prop_attr` ファイルを編集します。

```
domain-name Console::Access domain-name
Console:auths=solaris.vntsd.console-domain-name
```

このエントリは1行に収める必要があります。

- Oracle Solaris 11 OS の場合は、`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-name Console" \
'set desc="Access domain-name Console";
set auths=solaris.vntsd.console-domain-name'
```

- 4 権利プロファイルをユーザーに割り当てます。次のコマンドは、プロファイルをユーザーに割り当てます。

- Oracle Solaris 10 OS の場合は、権利プロファイルを割り当てます。

```
primary# usermod -P "All,Basic Solaris User,domain-name Console" username
All および Basic Solaris User プロファイルは必須です。
```

- Oracle Solaris 11 OS の場合は、権利プロファイルを割り当てます。

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

監査の有効化と使用

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、誰によって行われ、どのような影響があるかを示すログに保持されます。

この監査機能は、システムで実行されている次の Oracle Solaris OS のバージョンに基づいて有効または無効にすることができます。

- **Oracle Solaris 10 OS。** bsmconv および bsmunconv コマンドを使用します。 bsmconv(1M) および bsmunconv(1M) マニュアルページ、および『[System Administration Guide: Security Services](#)』のパート VII 「Auditing in Oracle Solaris」を参照してください。
- **Oracle Solaris 11 OS。** 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『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート VII 「Auditing in Oracle Solaris」を参照してください。

▼ 監査を有効にする方法

システムで Oracle Solaris 監査機能を構成して有効にする必要があります。 Oracle Solaris OS の監査機能は、制御ドメインで発生したアクションおよびイベントの履歴を検査するために使用されます。履歴は、何が、いつ、だれによって行なわれ、何が影響を受けたかのログに保存されます。 Oracle Solaris 11 の監査はデフォルトで有効になっていますが、いくつかの構成手順を実行する必要があります。

注-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 (Oracle Solaris 10) vs クラスを `/etc/security/audit_control` ファイルに追加します。
次の例の `/etc/security/audit_control` の部分は、vs クラスを指定する方法を示しています。

```
dir:/var/audit
flags:lo,vs
minfree:20
naflags:lo,na
```

- 3 (Oracle Solaris 10) 監査機能を有効にします。

- a. `bsmconv` コマンドを実行します。

```
# /etc/security/bsmconv
```

- b. システムをリブートします。

- 4 (Oracle Solaris 11) 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 クラスを含めることが重要です。

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。

ログアウトしたくない場合は、『[Oracle Solaris 11.1 Administration: Security Services](#)』の「[How to Update the Preselection Mask of Logged In Users](#)」を参照してください。

- 5 監査ソフトウェアが実行され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。

▼ 監査を無効にする方法

- 監査機能を無効にします。
 - Oracle Solaris 10 システムの監査機能を無効にします。

- a. **bsmunconv** コマンドを実行します。

```
# /etc/security/bsmunconv
Are you sure you want to continue? [y/n] y
This script is used to disable the Basic Security Module (BSM).
Shall we continue the reversion to a non-BSM system now? [y/n] y
bsmunconv: INFO: removing c2audit:audit_load from /etc/system.
bsmunconv: INFO: stopping the cron daemon.
```

```
The Basic Security Module has been disabled.
Reboot this system now to come up without BSM.
```

- b. システムをリブートします。

- Oracle Solaris 11 システムの監査機能を無効にします。

- a. **audit -t** コマンドを実行します。

```
# audit -t
```

- b. 監査ソフトウェアが実行されていないことを確認します。

```
# auditconfig -getcond
audit condition = noaudit
```

▼ 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。

▼ 監査ログをローテーションする方法

- **audit -n** コマンドを使用して、監査ログをローテーションします。
監査ログをローテーションさせて現在の監査ファイルを閉じ、現在の監査ディレクトリで別の新しいファイルを開きます。

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。

Oracle VM Server for SPARC Configuration Assistant を使用して、論理ドメインおよびサービスを構成することもできます。第 14 章「[Oracle VM Server for SPARC Configuration Assistant \(Oracle Solaris 10\)](#)」を参照してください。

この章では、次の項目について説明します。

- 63 ページの「出力メッセージ」
- 64 ページの「デフォルトのサービスの作成」
- 65 ページの「制御ドメインの初期構成」
- 67 ページの「Logical Domains を使用するためのレポート」
- 67 ページの「制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効化」
- 69 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」

出力メッセージ

Oracle VM Server for SPARC 2.0 リリース以降でリソースを制御ドメインに動的に構成できない場合は、最初に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、制御ドメインのリポ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 ドメインをリポートするまで、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。

- Oracle Solaris 10 の場合は、ネットワークアダプタドライバの仮想スイッチサービスを、制御ドメインに追加します。

```
primary# ldm add-vsw net-dev=net-driver vsw-service primary
```

たとえば、次のコマンドを使用して、ネットワークアダプタドライバ **nxge0** 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=nxge0 primary-vsw0 primary
```

- Oracle Solaris 11 の場合は、ネットワークアダプタドライバ **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=net-driver vsw-service primary
```

たとえば、次のコマンドを使用して、ネットワークアダプタドライバ `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
```

- 次は Oracle Solaris 10 OS のみに適用されるため、Oracle Solaris 11 システムでは実行しないでください。

このコマンドによって、仮想スイッチに MAC アドレスが自動的に割り当てられます。ldm add-vsw コマンドに、オプションとして独自の MAC アドレスを指定できます。ただし、この場合、指定した MAC アドレスが既存の MAC アドレスと競合していないことの確認は、ユーザーが責任を持って行います。

追加された仮想スイッチが、基本となる物理アダプタに代わりプライマリネットワークインタフェースとなる場合は、動的ホスト構成プロトコル (DHCP) サーバーによってドメインに同じ IP アドレスが割り当てられるように、仮想スイッチに物理アダプタの MAC アドレスを割り当てる必要があります。[67 ページの「制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効化」](#) を参照してください。

```
primary# ldm add-vsw mac-addr=2:04:4f:fb:9f:0d net-dev=nxge0 primary-vsw0 primary
```

- 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。

次のよう出力されるはずですが。

```
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 nxge0                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。

制御ドメインの初期構成を行うために、メモリーの動的再構成 (DR) を使用しないでください。リポートせずにメモリー DR を使用してこの構成を実行することは、可能ですが推奨されません。メモリー DR 手法は非常に長い時間を必要とし (リポートより時間がかかる)、失敗する可能性があります。代わりに、メモリー構成を変更する前に `ldm start-reconf` コマンドを使用して、遅延構成モードで制御ドメインを配置します。そのあと、すべての構成手順の実行後に制御ドメインをリポートできます。

▼ 制御ドメインを設定する方法

注-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。

- 1 制御ドメインに暗号化デバイスが割り当てられている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 システムなどの新しいプラットフォームには個別の暗号化装置が備わっていないため、これらのプラットフォームでは暗号化アクセラレータを割り当てる必要がありません。

```
primary# ldm list -o crypto primary
```

- 2 該当する場合は、暗号化リソースを制御ドメインに割り当てます。

次の例では、1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。これによって、残りの暗号化リソースをゲストドメインで使用できるようになります。

```
primary# ldm set-mau 1 primary
```

- 3 仮想 CPU 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、8つの仮想 CPU が制御ドメイン primary に割り当てられます。これにより、残りの仮想 CPU をゲストドメインで使用できるようになります。

```
primary# ldm set-vcpu 8 primary
```

- 4 制御ドメインの遅延再構成を開始します。

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

- 5 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、4G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。これにより、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 4G primary
```

- 6 論理ドメインのマシン構成をサービスプロセッサ (SP) に追加します。

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
```

- 7 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。

Logical Domains 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-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。

制御ドメインまたはサービルドメインとその他のドメイン間のネットワークの有効化



注意 - このセクションは、Oracle Solaris 10 システムのみに適用されます。Oracle Solaris 11 システムには `vsw` インタフェースを構成しないでください。

デフォルトでは、システムの制御ドメインとその他のドメイン間のネットワークは無効になっています。これを有効にするために、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ようにしてください。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上に構成されている限り、自動的に制御ドメインまたはサービルドメインと通信することができます。

注- この手順によって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ります。そのため、次の手順は制御ドメインのコンソールから実行してください。

▼ 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- 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、[167 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。

```
primary# 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。

```
primary# 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。

```
primary# 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。

```
primary# 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。

```
primary# 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。

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

```
primary# 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-vntsd を有効にする前に、制御ドメインにデフォルトのサービス vconscon (vcc) が作成されていることを確認してください。詳細は、64 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 **svcadm** コマンドを使用して、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 **svcs** コマンドを使用して、**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
```


◆◆◆ 第 5 章

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。

Oracle VM Server for SPARC Configuration Assistant を使用して、論理ドメインおよびサービスを構成することもできます。第 14 章「[Oracle VM Server for SPARC Configuration Assistant \(Oracle Solaris 10\)](#)」を参照してください。

この章では、次の項目について説明します。

- 71 ページの「[ゲストドメインの作成と起動](#)」
- 74 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を認識す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 8/11 OS を実行すると、すべての Oracle VM Server for SPARC 3.0 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 3.0 リリースノート](#)』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。

▼ 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。

たとえば、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。

たとえば、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
```

3 メモリーをゲストドメインに追加します。

たとえば、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。

たとえば、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イス名です。
- `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 - 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方法については、[132 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 [142 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。最初の例では、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2つめの例では、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- 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 は除外されません。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。

最初の例のコマンドは、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
```

2つめの例のコマンドは、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。『[Connecting Systems Using Fixed Network Configuration in Oracle Solaris 11.1](#)』および『[Connecting Systems Using Reactive Network Configuration in Oracle Solaris 11.1](#)』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 `volfd(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用します。

```
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    4     4G     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 のインストールメニューに従います。

▼ 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

```

- 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
```

次では、solarisdvd.iso を server1 サーバーに格納される Oracle Solaris ISO ファイルとして使用します。iso_vol@primary-vds0 はセカンダリボリューム、vdisk_iso は仮想ディスクです。lofiadm コマンドは、ブロック型デバイスを ISO イメージに接続します。

```
primary# lofiadm -a /net/server1/solarisdvd.iso
primary# ldm add-vdsdev /dev/lofi/1 iso_vol@primary-vds0
primary# ldm add-vdisk vdisk_iso iso_vol@primary-vds0 ldg1
```

- 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    4     4G      0.2%  22h 45m
...
VDS
NAME                VOLUME  OPTIONS  DEVICE
primary-vds0       vol1    /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
....
```

- 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 ..
```

- 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** のインストールメニューに従います。

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

注 - Oracle Solaris JumpStart 機能は Oracle Solaris 10 OS でのみ利用可能です。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 JumpStart to Oracle Solaris 11.1 Automated Installer](#)』を参照してください。

この手順では、ゲストドメインで Oracle Solaris JumpStart 機能を使用する方法を説明します。通常の JumpStart 手順を使用しますが、ゲストドメインの JumpStart プロファイルで使用するために異なるディスクデバイス名の形式で示します。『[Oracle Solaris 10 8/11 Installation Guide: Custom JumpStart and Advanced Installations](#)』を参照してください。

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。

- 使用する **JumpStart** プロファイルを修正して、この変更を反映してください。
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、124 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 8/11 Installation Guide: Custom JumpStart and Advanced 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 8/11 Installation Guide: Custom JumpStart and Advanced 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 ドメインの設定

この章では、I/O ドメインについて、および Logical Domains 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。

- 81 ページの「I/O ドメインの概要」
- 83 ページの「PCIe バスの割り当て」
- 89 ページの「PCIe エンドポイントデバイスの割り当て」
- 100 ページの「PCIe SR-IOV 仮想機能の使用」

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。I/O ドメインは、PCI EXPRESS (PCIe) バスまたは PCIe エンドポイントデバイスをドメインに割り当てることで作成できます。バスまたはデバイスをドメインに割り当てるには、`ldm add-io` コマンドを使用します。

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、他の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 ドメインの構成については、以下を参照してください。

- 83 ページの「PCIe バスの割り当て」
- 89 ページの「PCIe エンドポイントデバイスの割り当て」

注 - PCIe エンドポイントデバイスが構成されている I/O ドメインを移行することはできません。移行に関するその他の制限については、[第9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 帯域幅を効率的に利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、1 ギガビット Ethernet デバイスは、10 ギガビット Ethernet デバイスと比べて、より少ない CPU コアで全帯域幅を利用できる可能性があります。
- メモリ要件を順守します。メモリ要件は、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利用するには、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。

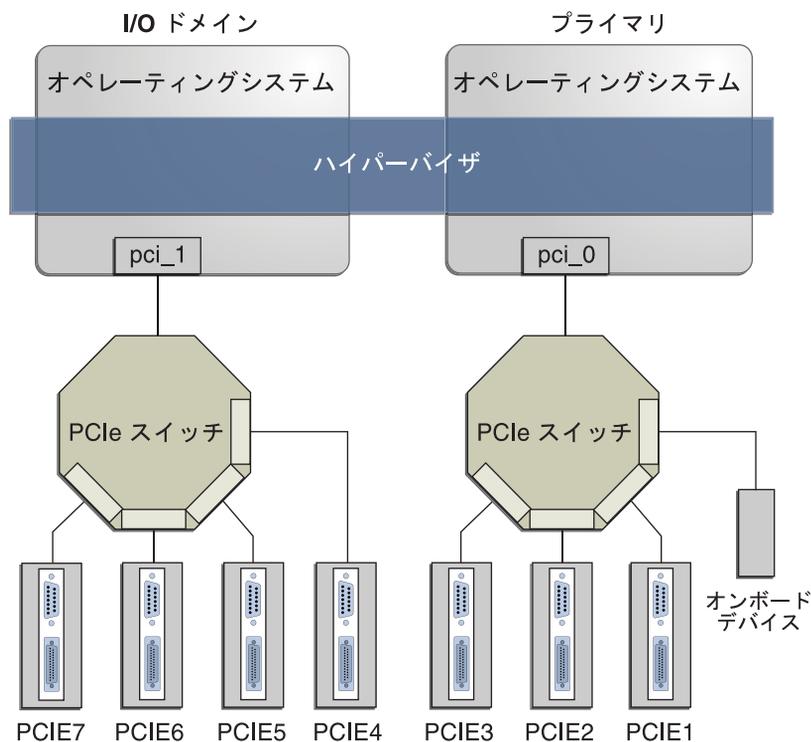
多数の仮想機能を作成して、十分な CPU およびメモリーリソースがないドメインに割り当てると、最適な構成にならない可能性があることに注意してください。

PCIeバスの割り当て

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、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、2つのPCIeバス(pci_0とpci_1)が存在するシステムを示しています。個々のバスが、別々のドメインに割り当てられています。このため、システムには2つのI/Oドメインが構成されます。

図 6-1 PCIeバスのI/Oドメインへの割り当て



PCIeバスで作成できるI/O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。たとえば、Oracle Sun SPARC Enterprise T5440サーバーを使用している場合、使用できるI/Oドメインは最大4つです。

注 – 一部の UltraSPARC サーバーには、PCIe バスは1つしか存在しません。このような場合、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で、I/O ドメインを作成できます。[89 ページ](#)の「[PCIe エンドポイントデバイスの割り当て](#)」を参照してください。システムにネットワークインタフェースユニット (NIU) が存在する場合、NIU をドメインに割り当てて I/O ドメインを作成することもできます。

PCIe バスを I/O 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 I/O ドメインに所有されます。そのバス上の PCIe エンドポイントデバイスを他のドメインに割り当てることはできません。primary ドメインに割り当てられている PCIe バス上の PCIe エンドポイントデバイスのみが、他のドメインに割り当て可能です。

Logical Domains 環境でサーバーが最初に構成される時、または factory-default 構成を使用しているとき、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。つまり、システムに構成されている I/O 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。

▼ PCIe バスの割り当てによって I/O ドメインを作成する方法

このプロシージャ例は、複数のバスが primary ドメインに所有されている初期構成から、新しい I/O 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。

最初に、primary ドメインのブートディスクを持つバスを保持する必要があります。それから、その他のバスを primary ドメインから削除してほかのドメインに割り当てます。



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。また、ドメインで使用されているネットワークポートなどのデバイスが接続されたバスを削除していないことを確認してください。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。

この例では、primaryドメインは1つのZFSプール(rpool(c0t1d0s0))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。

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
----                                -
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
```

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 ブートディスクが接続されている PCIe バスを確認します。

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 バスを確認します。

```
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
```

- 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。

■ Oracle Solaris 10.

```
primary# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
igb0: 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
```

■ Oracle Solaris 11.

```
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
```

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 ドメインを再構成する必要があります。

- 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、バス pci_1 が primary ドメインから削除されます。遅延再構成を開始する必要もあります。

```
primary# ldm start-reconf primary
primary# ldm remove-io pci_1 primary
```

- 7 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。

- 8 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
```

- 9 PCIeバスを追加するドメインを停止します。

ここでは、例として ldg1 ドメインを停止します。

```
primary# ldm stop ldg1
```

- 10 直接のアクセス権が必要なドメインに、使用可能なバスを追加します。

利用可能なバスは pci_1、ドメインは ldg1 です。

```
primary# ldm add-io pci_1 ldg1
```

- 11 ドメインを再起動して、変更を有効にします。

次のコマンドでは、ldg1 ドメインを再起動します。

```
primary# ldm start ldg1
```

- 12 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
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
pci_0	BUS	pci_0	primary	
pci_1	BUS	pci_1	ldg1	
/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	ldg1	EMP
/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	

この出力では、PCIeバス pci_0 とそのデバイスが 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_1 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 エンドポイントデバイスの割り当て

Oracle VM Server for SPARC 2.0 リリースと Oracle Solaris 10 9/10 OS 以降では、個別の 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の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

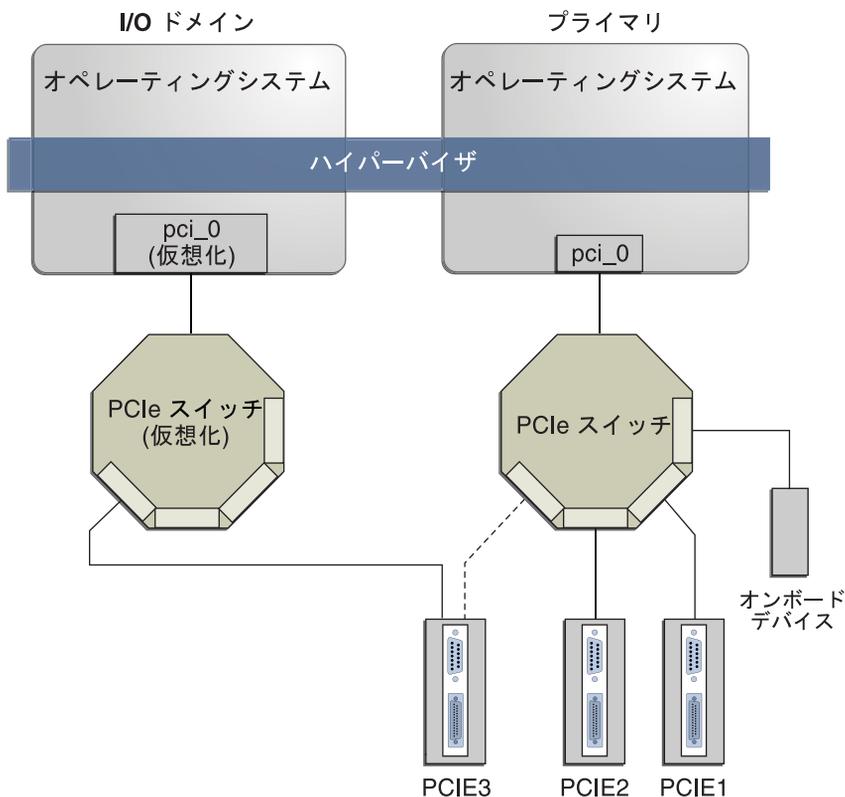
次の図は、PCIe エンドポイントデバイス PCIe3 が I/O ドメインに割り当てられている状態を示しています。I/O ドメインのバス `pci_0` とスイッチは、いずれも仮想のものです。PCIe3 エンドポイントデバイスには、`primary` ドメインではアクセスできなくなっています。

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非常によく似ています。`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 rm-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後、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、[94 ページの「PCIe ハードウェアの変更」](#)を参照してください。

図 6-2 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。『Oracle VM Server for SPARC 3.0 リリースノート』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。

DIO 機能についてのいくつかの重要な詳細を以下に示します。

- 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。『Oracle VM Server for SPARC 3.0 リリースノート』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。
- primary 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 機能によって他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、primary 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。93 ページの「[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、バスを管理する。
 - 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

DIO 機能を正しく使用するには、適切なソフトウェアを実行し、DIO 機能によってサポートされている PCIe カードのみを I/O ドメインに割り当てる必要があります。ハードウェアおよびソフトウェアの要件については、『Oracle VM Server for SPARC 3.0 リリースノート』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-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、primary 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 primary ドメインの変更が最大限になり、primary 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

以下の制限を回避する方法については、92 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。

- 他のドメインに 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。

PCIe エンドポイントデバイス構成の計画

primary 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。primary ドメインのリブートは、primary ドメイン自体で利用可能なサービスだけでなく、PCIe エンドポイントデバイスが割り当てられている I/O ドメインにも影響を及ぼします。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 primary 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、95 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、一般に以下のような手順が必要です。

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 primary 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 rm-io` コマンドを使用します。`rm-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-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、`primary` ドメインを1回だけリブートする必要があります。

- 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。
- 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。
- 削除したPCIe エンドポイントデバイスが、`primary` 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。
- 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。
- ゲストドメインに対して、PCIe エンドポイントデバイスの追加または削除を行います。
`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。
- (オプション)PCIe ハードウェアに変更を加えます。
94 ページの「[PCIe ハードウェアの変更](#)」を参照してください。

ルートドメインのリブート

`primary` 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。`primary` ドメインはアクティブで、DIO 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。`primary` 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 `primary` 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。`primary` 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。

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。

- primary 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、primary ドメインのシャットダウン、停止、またはリブートをを行う前に、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。計画を行うときに、この情報を使用します。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。
- primary 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 primary 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。

```
# ldm set-domain failure-policy=reset primary
# ldm set-domain master=primary ldom
```

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 rm-io` コマンドを使用して、まず primary 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、primary ドメインに割り当てられるときは、作業は必要ありません。
- 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 primary 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 rm-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 `primary` ドメインに再割り当てします。そのあとで、`primary` 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 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、[92 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 - 今回のリリースでは、`DefaultFixedNCP` 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ことをお勧めします。

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せん。
- 「自動」。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せん。

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。prttdiag -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-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。

```
# 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。
84 ページの「PCIe バスの割り当てによって I/O ドメインを作成する方法」の段階 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。
84 ページの「PCIe バスの割り当てによって I/O ドメインを作成する方法」の段階 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。
84 ページの「PCIe バスの割り当てによって I/O ドメインを作成する方法」の段階 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。

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。

```
# 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。

```
# 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。

```
# reboot -- -r
```

- 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。

```
# 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		OCC
/SYS/MB/PCIE3	PCIE	pci_1		OCC
/SYS/MB/PCIE5	PCIE	pci_1		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 - `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。

```
# ldm add-io /SYS/MB/PCIE3 ldg1
```

b. `ldg1` ドメインをバインドし、起動します。

```
# ldm bind ldg1
# 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。

■ Oracle Solaris 10。次のコマンドを実行します：

```
# dladm show-dev
nxge0      link: unknown   speed: 0        Mbps        duplex: unknown
nxge1      link: unknown   speed: 0        Mbps        duplex: unknown
nxge2      link: unknown   speed: 0        Mbps        duplex: unknown
nxge3      link: unknown   speed: 0        Mbps        duplex: unknown
```

■ Oracle Solaris 11。次のコマンドを実行します：

```
# 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
```

PCIe SR-IOV 仮想機能の使用

Oracle VM Server for SPARC 2.2 以降のリリースでは、Peripheral Component Interconnect Express (PCIe) シングルルート I/O 仮想化 (SR-IOV) 機能のサポートが追加されています。

SR-IOV の概要

SR-IOV の実装は、PCI-SIG に定義されているバージョン 1.1 の標準に基づきます。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

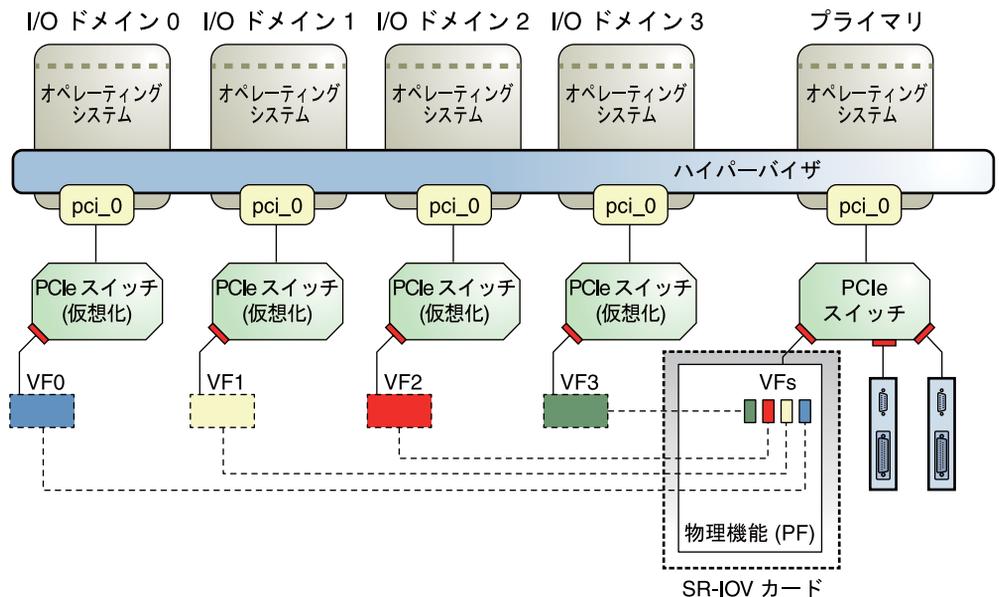
イティブのパフォーマンスに匹敵する I/O パフォーマンスを達成するためにハードウェアに実装されます。SR-IOV 仕様は、新しいデバイスを作成し、それにより仮想マシンを I/O デバイスに直接接続できるという、新しい標準を定義しています。

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 6-3 I/O ドメインでの仮想機能と物理機能の使用



SR-IOV には次の機能の種類があります。

- 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 64,000 の仮想機能に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。

SR-IOV 機能を持つデバイスに関連する利点は次のとおりです。

- 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。
 - 節電
 - アダプタ数の削減
 - 配線の削減
 - スイッチポートの削減

SR-IOV のハードウェア要件とソフトウェア要件

Oracle VM Server for SPARC 2.2 以降のリリースでは、PCIe SR-IOV 機能のサポートが追加されています。必要なハードウェア、ソフトウェア、およびファームウェアのバージョンの詳細は、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』の「[PCIe SR-IOV のハードウェアおよびソフトウェア要件](#)」を参照してください。

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。

ルートドメインをリポートすると 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ポートが最小限になるように、十分に計画してください。

SR-IOV 機能の現在の制限事項

注-リポートを最小限に抑えるには、同じ遅延再構成の中で複数の操作を実行します。

今回のリリースでは、SR-IOV 機能に次の制限事項があります。

- 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。
- Ethernet SR-IOV カードのみがサポートされます。
- SR-IOV 機能は、primary ドメインに取り付けられた SR-IOV カードに対してのみ有効です。PCIe バスの割り当てまたは直接 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。
- 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。

PCIe SR-IOV 仮想機能の使用の計画

仮想機能の作成と破棄の前に、前もって計画して、構成で使用する仮想機能を決定してください。仮想機能を作成および破棄するときは、primary ドメインをリポートする必要があります。このようなリポートは、PCIe エンドポイントまたは SR-IOV 仮想機能のいずれかが構成されているすべての I/O ドメインに悪影響を及ぼします。そのため、primary ドメインのリポート回数は最小限に抑えることが重要です。さまざまな SR-IOV デバイスから、現在の構成と将来の構成のニーズを満たすために必要な仮想機能の数を決定します。

I/O ドメインの詳細は、82 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決定します。

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。114 ページの「SR-IOV の詳細なトピック」を参照してください。

- 2.

ルートドメインで遅延再構成を開始します。

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

3. PCIe バスに対する I/O 仮想化操作を有効にします。

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

4. 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。

次のコマンドを使用して、仮想機能を作成します。

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

`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`s、`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`s、および `mtu` のネットワーク固有のプロパティは、次のように変更できます。

- 仮想機能が `primary` ドメインに割り当てられている場合: 遅延再構成を開始する必要があります。
- 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停止しているときに行な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。

遅延再構成時には、複数の仮想機能を作成し、primary ドメインのリブートを 1 回実行するだけで、変更を有効にすることができます。仮想機能を作成するたびに primary ドメインをリブートする必要はありません。

特定の SR-IOV 物理機能が多数の仮想機能をサポートする場合があります。必要な仮想機能だけを作成してください。推奨される構成の最大数については、114 ページの「SR-IOV の詳細なトピック」を参照してください。

5. `ldm add-config` コマンドを使用して、構成を SP に保存します。
6. ルートドメインをリブートして、仮想機能を作成します。
7. `ldm add-io` コマンドを使用して仮想機能をアクティブなドメインに割り当てる前に、そのドメインを停止する必要があります。すべての I/O ドメインにまとめて変更を加えることで、I/O ドメインのダウンタイムを最小限に抑えます。この方法で、このような構成を設定するために必要な primary ドメインのリブート回数を減らすことができます。
8. I/O ドメインをブートし、仮想機能をほかのネットワークデバイスと同じように構成します。
仮想機能の制限事項については、114 ページの「SR-IOV の詳細なトピック」を参照してください。

仮想機能の作成、変更、および破棄

このセクションでは、仮想機能の作成、変更、および破棄の方法を説明します。

▼ 仮想機能を作成する方法

- 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 制御ドメインの遅延再構成を開始します。

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

- 3 物理機能を持つバスでまだ I/O 仮想化が有効になっていない場合は、`iovs-on` のみを設定して有効にします。

```
primary# ldm set-io iovs-on bus
```

4 物理機能から仮想機能を作成します。

```
primary# 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 - MAC アドレスはネットワークデバイスに対して自動的に割り当てられます。

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、仮名を使用することをお勧めします。

例 6-1 仮想機能の作成

次の例は、物理機能 /SYS/MB/NET0/IOVNET.PF0 についての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
```

次の例は、仮想機能の作成方法を示しています。

- 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。

```
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
```

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

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

- この例では、2つの代替MACアドレスがある仮想機能を作成します。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
```

▼ 仮想機能を変更する方法

`ldm set-io bus` 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、114ページの「SR-IOVの詳細なトピック」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`、および `mtu`
これらの仮想機能プロパティを変更するには、まず仮想機能を所有しているドメインを停止し、遅延再構成を開始します。変更を有効にするには、ドメインをリブートする必要があります。
- `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。これを行うと、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るため、注意してください。つまり、`pvid` プロパティを設定すると、透過的な 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。変更を有効にするには、遅延再構成を開始し、`primary` ドメインをリポートする必要があります。デバイス固有のプロパティの詳細については、114 ページの「SR-IOV の詳細なトピック」を参照してください。

- 仮想機能を変更します。

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

例 6-2 仮想機能の変更

次の例は、`ldm set-io` コマンドで仮想機能にプロパティを設定する方法を示しています。

- 次の例では、指定された仮想機能 `/SYS/MB/NET0/IOVNET.PF0.VF0` を、VLAN ID 2、3、および 4 の一部になるように変更します。

```
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ります。ことに注意してください。

注- 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.

▼ 仮想機能を破棄する方法

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せん。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。変更を有効にするには、遅延再構成を開始し、ドメインをリブートする必要があります。

- 1 ルートドメインで遅延再構成を開始します。

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

- 2 仮想機能を破棄します。

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

例 6-3 仮想機能の破棄

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

```
primary# ldm destroy-vf /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 ドメイン上の仮想機能の追加および削除

▼ 仮想機能を I/O ドメインに追加する方法

次のコマンドは、仮想機能を論理ドメインに追加します。

```
ldm add-io vf-name ldom
```

`vf-name` は、仮想機能の仮名またはパス名です。仮名を使用することをお勧めします。`ldom` 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。primary ドメインを指定する場合は、遅延再構成を開始する必要があります。

- 仮想機能を追加します。

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

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。

例 6-4 仮想機能の追加

次の例は、/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ています。正常に実行するには、指定したドメインがアクティブでないか停止している必要があります。そのドメインが primary 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。

```
primary# ldm add-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 ドメインに追加されます。ldg1 がすでにバインドされている (またはあとでバインドされる) 場合は、ドメインを起動することができ、ゲスト OS が I/O 操作に追加された仮想機能を使用することができません。

▼ 仮想機能を I/O ドメインから削除する方法

次のコマンドは、SR-IOV 仮想機能を論理ドメインから削除します。

```
ldm remove-io vf-name ldom
```

vf-name は、仮想機能の仮名またはパス名です。デバイスの仮名を使用することをお勧めします。ldom 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。primary ドメインを指定する場合は、遅延再構成を開始する必要があります。

注- 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。

- 仮想機能を削除します。

```
primary# ldm rm-io vf-name ldom
```

例 6-5 仮想機能の削除

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

```
primary# ldm rm-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。

仮想機能を持つドメインが primary ドメインの場合は、遅延再構成を開始する必要があります。

SR-IOV: ルートドメインのリブート

primary ドメインをリブートするときは、慎重に行なってください。[93 ページ](#)の「ルートドメインのリブート」を参照してください。I/O ドメインの PCIe スロットと同様に、このセクションで説明している問題は、I/O ドメインに割り当てられている仮想機能にも関連があります。

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

primary ドメインのリブート回数を最小限に抑えて、ダウンタイムを最小限に抑えるように事前に計画します。

- 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 共有する物理機能についての詳細情報を入手します。

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

- 3 ルートドメインで遅延再構成を開始します。

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

- 4 PCIe バスに対する I/O 仮想化を有効にします。

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

- 5 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。これらのコマンドをバッチとして実行すると、primary ドメインを1回リブートするだけで済みます。

- 6 PCIe エンドポイントデバイスまたは仮想機能が割り当てられているすべての I/O ドメインを停止します。

```
primary# ldm stop ldom
```

- 7 ルートドメインをリブートします。

```
primary# reboot
```

- 8 primary 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
```

- 9 仮想機能をドメインに割り当てます。

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

- 10 ドメインをバインドし、起動します。

```
primary# ldm bind ldom
primary# ldm start ldom
```

- 11 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 6-6 SR-IOV 仮想機能を割り当てることによる I/O ドメインの作成

次の例は、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。

次の ldm list-io 出力は、/SYS/MB/NET0/IOVNET.PF0 物理機能が使用可能であ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
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、作成できる仮想機能の最大数を含む、`/SYS/MB/NET0/IOVNET.PF0` 物理機能に関する追加の詳細を表示します。

```

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

```

次のコマンドは、`/SYS/MB/NET0/IOVNET.PF0` 物理機能に対して `/SYS/MB/NET0/IOVNET.PF0.VF0` という仮想機能を作成します。

```

primary# ldm start-reconf primary
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

```

ldg1 I/O ドメインには DIO 機能を使用して作成された PCIe エンドポイントデバイスがあるため、次のように、ldg1 ドメインを停止し、primary ドメインをリブートする必要があります。

```

primary# ldm stop ldg1
primary# reboot

```

次のコマンドは、新しい `/SYS/MB/NET0/IOVNET.PF0.VF0` 仮想機能が存在することを確認します。

```

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  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		

次のコマンドは、/SYS/MB/NET0/IOVNET.PF0.VF0 仮想機能を ldg1 ドメインに割り当てます。

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

次のコマンドは、ldg1 ドメインをバインドして再起動します。

```
primary# ldm bind ldg1
primary# ldm start ldg1
```

次のコマンドは、仮想機能が使用可能であることを確認します。

```
guest# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
net0          Ethernet      up       0      unknown vnet0
net1          Ethernet      up      1000   full    igbvf0
```

SR-IOVの詳細なトピック

このセクションでは、PCIe SR-IOV 対応の I/O デバイスの使用時に生じる、いくつかの詳細なトピックについて説明します。

- **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、ほかの種類の 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 - 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。

- サポートされる I/O ドメインおよび仮想機能の最大数。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 15 まで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。そのため、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。SR-IOV に関連する問題については、『Oracle VM Server for SPARC 3.0 リリースノート』を参照してください。

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。各プロパティに、名前、簡単な説明、デフォルト値、最大値、および次のフラグの 1 つまたは複数があります。

- 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には、遅延再構成を開始する必要があります。

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
```

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 ギガビット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作成することもできます。

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。そのため、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、I/O ドメインで Oracle Solaris 11 OS を起動してブートします。この例では、ldg1 が I/O ドメインです。

```
primary# ldm start ldg1
```

次のコマンドは、ゲストドメインで Oracle Solaris 11 の dladm コマンドを使用して、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 show-phys
LINK          MEDIA          STATE          SPEED  DUPLEX  DEVICE
net0          Ethernet      up             0      unknown vnet0
net1          Ethernet      up            1000   full    igbvf0
guest# dladm create-vnic -l net1 vnic0
guest# dladm create-vnic -l net1 vnic1
guest# dladm create-vnic -l net1 vnic2
guest# dladm create-vnic -l net1 vnic3
guest# dladm show-link
LINK          CLASS  MTU  STATE  OVER
net0          phys   1500 up     --
net1          phys   1500 up     --
vnic0         vnic   1500 up     net1
vnic1         vnic   1500 up     net1
vnic2         vnic   1500 up     net1
vnic3         vnic   1500 up     net1
```


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。

- 119 ページの「仮想ディスクの概要」
- 120 ページの「仮想ディスクの管理」
- 123 ページの「仮想ディスクの識別子とデバイス名」
- 124 ページの「仮想ディスクの表示」
- 125 ページの「仮想ディスクバックエンドオプション」
- 127 ページの「仮想ディスクバックエンド」
- 134 ページの「仮想ディスクマルチパスの構成」
- 137 ページの「CD、DVD および ISO イメージ」
- 141 ページの「仮想ディスクのタイムアウト」
- 141 ページの「仮想ディスクおよび SCSI」
- 142 ページの「仮想ディスクおよび format コマンド」
- 142 ページの「仮想ディスクと ZFS の使用」
- 147 ページの「Logical Domains 環境でのボリュームマネージャーの使用」

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 の終端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。

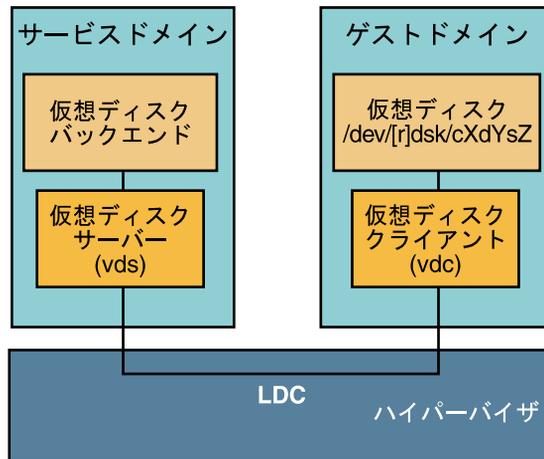
仮想ディスクバックエンドは、物理的でも論理的でもかまいません。物理デバイスには、次のものを含めることができます。

- 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、UFS などのファイルシステムのファイル
- 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 7-1 Logical Domains による仮想ディスク



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、[125 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、[141 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 ldom
```

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 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。123 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-バックエンドは、ゲストドメイン (*ldom*) 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

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。



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 次のコマンドを使用して、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 次のコマンドを使用して、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、[125 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、次のコマンドを使用して仮想ディスクオプションを変更できます。

```
# 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、[125 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、次のコマンドを使用して仮想ディスクのタイムアウトを変更できます。

```
# ldm set-vdisk timeout=seconds disk-name ldom
```

▼ 仮想ディスクを削除する方法

- 1 次のコマンドを使用して、ゲストドメインから仮想ディスクを削除します。

```
# ldm rm-vdisk disk-name ldom
```

- 2 次のコマンドを使用して、サービスドメインからの対応するバックエンドのエクスポートを停止します。

```
# ldm rm-vdsdev volume-name@service-name
```

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 ldom
```

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。

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、8つのスライス (s0 - s7) を持つ通常のディスクとしてドメインに表示されます。このようなディスクは、`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。ただし、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、8つのスライス (s0 - s7) を持つ通常のディスクとしてドメインに表示されます。ただし、使用できるのは1番目のスライス (s0) のみです。このようなディスクは、`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 - 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`)。このようなディスクは、`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、サービスドメ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、サービスドメ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。これによって、サービスドメ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 - 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は機能する場合と機能しない場合があります。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、サービスドメ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、148 ページの「Solaris Volume Manager での仮想ディスクの使用」を参照してください。
- Veritas Volume Manager (VxVM) がサービスドメ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、149 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、121 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、サービスド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。これは、Oracle Solaris 10 5/08 OS リリースから導入された新しい動作です。Oracle Solaris 10 5/08 OS より前のリリースでは、ディスクバックエンドは常に排他的に開かれ、バックエンドを排他的でない状態で開くことはできませんでした。

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、127 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ド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
```

- 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`
- 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せん。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、サービス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。このようにしないと、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。

▼ 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

- 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 `c1t57d0` 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。
- 3 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 (`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。

```
primary# ldm add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

```
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。このようにしないと、ディスク全体がエクスポートされます。

```
# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

ファイルおよびボリューム

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 - Oracle Solaris 5/08 OS より前のリリースでは、空のファイルが仮想ディスクとしてエクスポートされると、システムによってデフォルトのディスクラベルが書き込まれ、デフォルトのパーティションが作成されていました。Oracle Solaris 5/08 OS リリースではこの処理は行われなくなったため、ゲストドメインで `format(1M)` を実行してパーティションを作成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/c0d5s*  
/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

リユ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ています。

- 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

Oracle Solaris 10 5/08 OS より前のリリースでは、`slice` オプションがなく、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ていました。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である場合に、そのシステムを Oracle Solaris 10 5/08 OS にアップグレードすると、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 3.0 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。このコマンドの詳細は、[ldm\(1M\)](#) マニュアルページを参照してください。
- 次の行を、サービスマインの `/etc/system` ファイルに追加します。

```
set vds:vd_volume_force_slice = 1
```

注- 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	1つのスライスディスク ²
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 ³	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
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

¹ ディスク全体をエクスポートします。

² スライス2のみをエクスポートします。

³ 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

ループバックファイル (lofi) ドライバを使用すると、ファイルを仮想ディスクとしてエクスポートできます。ただし、これを行うと別のドライバ層が追加され、仮想ディスクのパフォーマンスに影響を及ぼします。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。[129 ページの「ファイルおよびボリューム」](#) 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

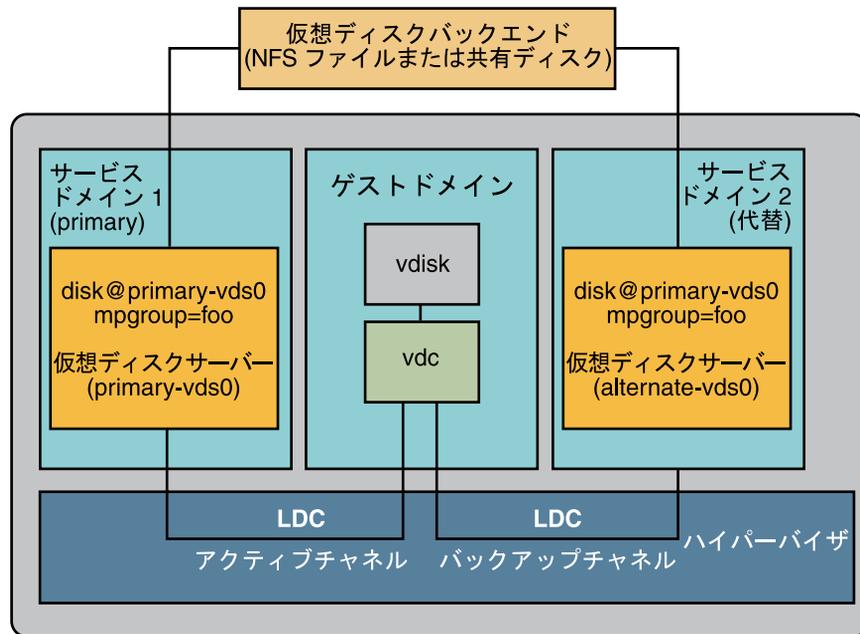
仮想ディスクマルチパスを使用すると、ゲストドメインに仮想ディスクを構成して、複数のパスからバックエンドストレージにアクセスできます。それらのパスは、ディスク LUN などの同一のバックエンドストレージにアクセスできる各種サービスドメインを通過します。この機能では、サービスドメインのいずれかがダウンしても、ゲストドメイン内の仮想ディスクをアクセス可能なままにしておくことができます。たとえば、仮想ディスクマルチパス構成を設定して、ネットワークファイルシステム (Network File System、NFS) サーバーのファイルにアクセスすることがあります。また、この構成を使用して、複数のサービス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 - Oracle VM Server for SPARC 2.0 リリース以降、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、136 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、foo 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、第一サービスドメインと代替サービスドメインの 2 つからアクセスできます。

図 7-2 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および仮想ディスクのタイムアウト

仮想ディスクマルチパスを使用すると、バックエンドが現在アクティブなパスによってアクセス不能になった場合に、バックエンドへのアクセスに使用されるパスが自動的に変更されます。このパスの変更は、仮想ディスクの timeout プロパティの値とは無関係に行なわれます。

仮想ディスクの `timeout` プロパティは、I/O を処理できるサービスドメインがないときに、I/O が失敗するまでの時間を指定します。このタイムアウトは、仮想ディスクマルチパスを使用する仮想ディスクを含め、すべての仮想ディスクに適用されます。

結果として、仮想ディスクマルチパスが構成されているときに仮想ディスクのタイムアウトを設定すると、マルチパスが正しく機能しない可能性があります。これは特に、小さいタイムアウト値の場合です。そのため、マルチパスグループの一部である仮想ディスクに仮想ディスクのタイムアウトを設定することはお勧めできません。

詳細については、[141 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。

▼ 仮想ディスクマルチパスを構成する方法

- 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。

```
# ldm add-vdsdev mpgroup=foo backend-path1 volume@primary-vds0
```

`backend_path1` は、`primary` ドメインから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。

```
# ldm add-vdsdev mpgroup=foo backend-path2 volume@alternate-vds0
```

`backend_path2` は、`alternate` サービスドメインから仮想ディスクバックエンドへのパスです。

注 - `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。

```
# ldm add-vdisk disk-name volume@primary-vds0 ldom
```

注 - 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。

参考 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サービスドメイン(この例ではprimary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。

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 - 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。エクスポートできるのは、CD または DVD ドライブ内の CD または DVD のみです。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され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、138 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 - 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 を要求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、要求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

- 1 サービスドメインから、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 vold
STATE          STIME          FMRI
online         12:28:12      svc:/system/filesystem/voldfs:default
```

- 2 次のいずれかの手順を実行します。

- ボリューム管理デーモンが動作中またはオンラインでない場合は、手順 3 に進みます。
- 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` ファイルを編集して、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/rdisk/clt0d0s2                | 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。

次の例は、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
```

参考 CD または DVD の複数回のエクスポート

CD または DVD は複数回エクスポートし、異なる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、[121 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

ここでは、`primary` ドメインから **ISO** イメージをエクスポートし、それを使用してゲストドメインをインストールする手順について説明します。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。

```
# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    4     4G     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。

```
# 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。

```
# 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。

```
# 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。

```
# ldm start-domain ldom1
LDom ldom1 started
# ldm list
NAME           STATE   FLAGS   CONS   VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv   SP     4     4G      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。

```
# 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は、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、入出力処理が失敗してエラーが報告される方が望ましい場合があります。現在は、仮想ディスクごとに接続タイムアウト時間を設定することが可能になり、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。

このタイムアウトは、次のいずれかを実行すると設定できます。

- `ldm add-vdisk` コマンドを使用します。

```
ldm add-vdisk timeout=seconds disk-name volume-name@service-name ldom
```

- `ldm set-vdisk` コマンドを使用します。

```
ldm set-vdisk timeout=seconds disk-name ldom
```

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)。

また、ゲストドメインの `/etc/system` ファイルに次の行を追加すると、タイムアウトを設定できます。

```
set vdc:vdc_timeout=seconds
```

注 - 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。

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。

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せん。

注-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。そうでないと、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 では次のことを実行できます。

- 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複製を使用する

ZFS の使用法の詳細については、『[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Oracle Solaris OSバージョンのインストールの種類に合わせます。

ZFSによるディスクイメージの格納例

次の手順を実行します。

1. ZFSボリュームまたはZFSファイルに20Gバイトのイメージを作成します。
2. ZFSボリュームまたはZFSファイルを仮想ディスクとしてエクスポートします。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。
3. エクスポートされたZFSボリュームまたはZFSファイルをゲストドメインに割り当てます。

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。

▼ ZFS ボリュームを使用してディスクイメージを作成する方法

- たとえば、ZFS ボリュームに20Gバイトのディスクイメージを作成します。

```
primary# zfs create -V 20gb ldmpool/ldg1/disk0
```

▼ ZFS ファイルを使用してディスクイメージを作成する方法

- たとえば、ZFS ボリュームに20Gバイトのディスクイメージを作成します。

```
primary# zfs create ldmpool/ldg1/disk0
primary# mkfile 20g /ldmpool/ldg1/disk0/file
```

▼ ZFS ボリュームをエクスポートする方法

- 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
```

▼ ZFS ファイルをエクスポートする方法

- 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_disk0@primary-vds0
```

▼ ZFS ボリュームまたはZFS ファイルをゲストドメインに割り当てる方法

- ZFS ボリュームまたはZFS ファイルをゲストドメイン(次の例ではldg1)に割り当てます。

```
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 スナップショット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。ゲストドメインで確実にディスクが使用中ではない状態にするには、いくつかの方法があります。次のいずれかの手順を実行します。

- 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、また、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のマウントを解除し、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。

▼ ディスクイメージのスナップショットを作成する方法

- たとえば、`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 複製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、`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-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 `sys-unconfig` コマンドを実行します。
- 3 `sys-unconfig` コマンドが完了すると、このドメインは停止します。
- 4 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。

- 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。

- 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。

Logical Domains 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、Logical Domains 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での仮想ディスクの使用

ZFS (Zettabyte File System)、Solaris Volume Manager、または Veritas Volume Manager (VxVM) は、サービスド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 - 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 Volume Manager での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ペアを使用した 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ペア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ペアに交換して、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。

```
# 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 Veritas Volume Manager (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
```

コマンド `vxdisk list` で出力される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。次に例を示します。

```
# 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。

仮想ディスクでの 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 Volume Manager** 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、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。

- 152 ページの「仮想ネットワークの概要」
- 152 ページの「Oracle Solaris 10 ネットワークの概要」
- 154 ページの「Oracle Solaris 11 ネットワークの概要」
- 157 ページの「仮想スイッチ」
- 158 ページの「仮想ネットワークデバイス」
- 160 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」
- 163 ページの「自動または手動による MAC アドレスの割り当て」
- 166 ページの「Logical Domains でのネットワークアダプタの使用」
- 167 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」
- 171 ページの「Logical Domains 環境での IPMP の構成」
- 179 ページの「VLAN のタグ付けの使用」
- 183 ページの「NIU ハイブリッド I/O の使用」
- 187 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」
- 188 ページの「ジャンボフレームの構成」
- 193 ページの「Oracle Solaris 11 のネットワーク固有の機能の相違点」

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、152 ページの「Oracle Solaris 10 ネットワークの概要」、154 ページの「Oracle Solaris 11 ネットワークの概要」、193 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 で大きく異なります。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。

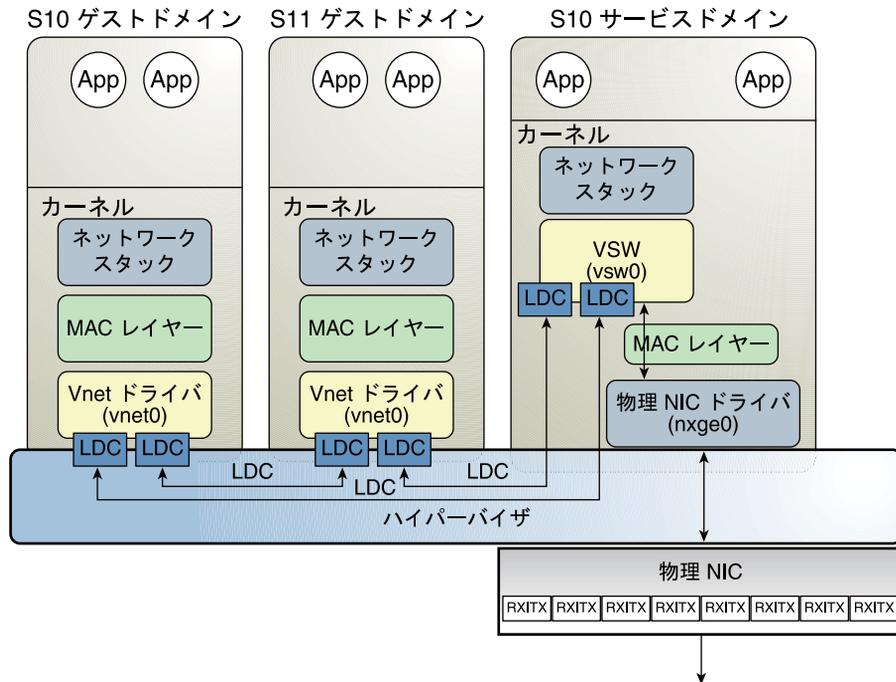
注 - Oracle Solaris 10 のネットワークは、ドメインまたはシステムで同じように動作します。Oracle Solaris 11 ネットワークについても同様です。Oracle Solaris OS のネットワークの詳細は、[Oracle Solaris 10 Documentation](#) および [Oracle Solaris 11.1 Documentation](#) を参照してください。

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、[154 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 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 8-1 Oracle Solaris 10 OS の Oracle VM Server for SPARC ネットワークの概要



以下に、`nxge0`、`vsw0`、`vnet0`などのインタフェース名を示している前の図について説明します。これらのインタフェース名はOracle Solaris 10 OSにのみ適用されます。

- 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。

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 11.1 Documentation](#) の Oracle Solaris 11 ネットワークドキュメントで説明しています。

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 (vsw) 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。

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断するには、dladm show-phys の出力で vsw を検索します。

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。

注 - 仮想スイッチ (vsw) または仮想ネットワークデバイス (vnetn) での VNIC の作成はサポートされていません。『Oracle VM Server for SPARC 3.0 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 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。

注 - 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することをお勧めします。

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。
- 「自動」。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。

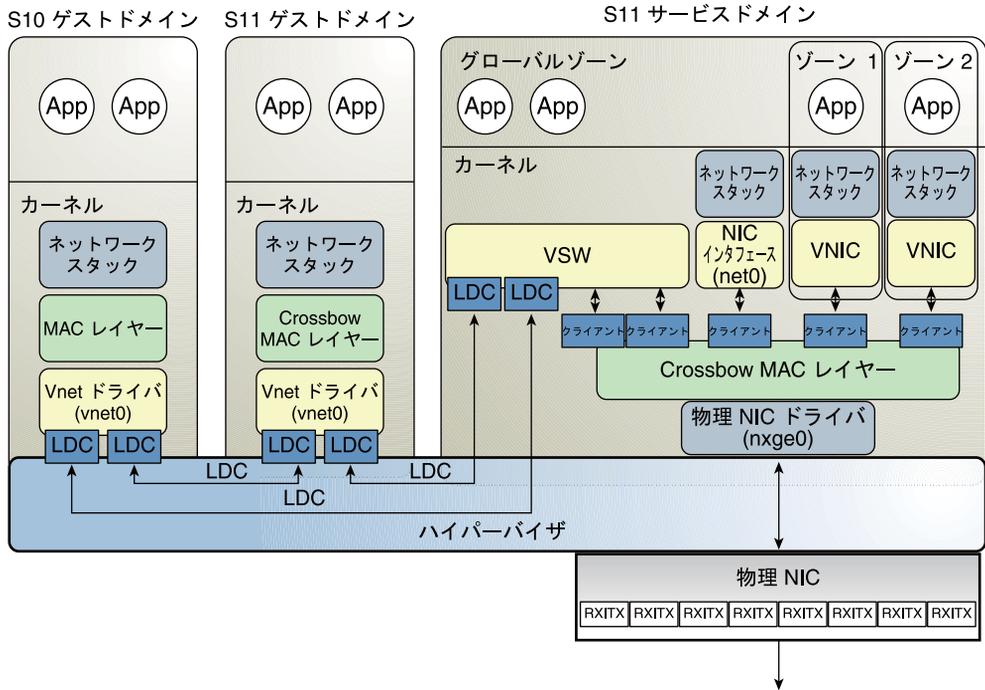
netadm list コマンドを使用して、DefaultFixed NCP が有効になっていることを確認します。『Oracle Solaris Administration: Network Interfaces and Network Virtualization』の第7章「Using Datalink and Interface Configuration Commands on Profiles」を参照してください。

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。制御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、仮想ネットワークデバイスが同じ仮想スイッチに接続されているゲストドメインと通信します。
- 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。

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 8-2 Oracle Solaris 11 OS の Oracle VM Server for SPARC ネットワークの概要



以下は、前の図について説明しており、`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。

- サービスドメイン内の仮想スイッチは、ゲストドメインに接続されます。これにより、ゲストドメイン間で相互に通信することができます。
- 仮想スイッチは物理ネットワークデバイス `nxge0` にも接続されています。これにより、ゲストドメインは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。そのため、Oracle Solaris 11 MAC レイヤーのネットワークの機能拡張により、Oracle Solaris 11 サービスドメインで、`vsw` をネットワークインタフェースとして構成する必要はありません。

- 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せん。
- 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。

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。仮想スイッチはネットワークインタフェース `vswi` も備えています。このインタフェースによって、サービスドメインは、仮想スイッチに接続されたほかのドメインと通信できます。仮想スイッチは通常のネットワークインタフェースと同様に使用でき、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 - 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、67 ページの「[制御ドメインまたはサービスドメインとその他のドメイン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。

次の例は、物理ネットワークアダプタに仮想スイッチを作成する方法を説明しています。

- **Oracle Solaris 10:** 次のコマンドは、`nxge0` という物理ネットワークアダプタに仮想スイッチを作成します。

```
primary# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

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詳細については、67 ページの「[制御ドメインまたはサービスドメインとその他のドメイン間のネットワークの有効化](#)」を参照してください。

- **Oracle Solaris 11:** 次のコマンドは、net0 という物理ネットワークアダプタに仮想スイッチを作成します。

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

仮想ネットワークデバイス

仮想ネットワーク (vnet) 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 - 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 8-1 および 図 8-2 で Oracle Solaris 10 および Oracle Solaris 11 の Oracle VM Server for SPARC ネットワークに関する情報をそれぞれ参照してください。

Inter-Vnet LDC チャネル

Oracle VM Server for SPARC 2.1 リリースまでの 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、1つの仮想スイッチデバイス内で仮想ネットワークデバイスの数が増加すると、inter-vnet 通信に必要な LCD チャネルの数も急激に増加します。

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` チャン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 チャン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

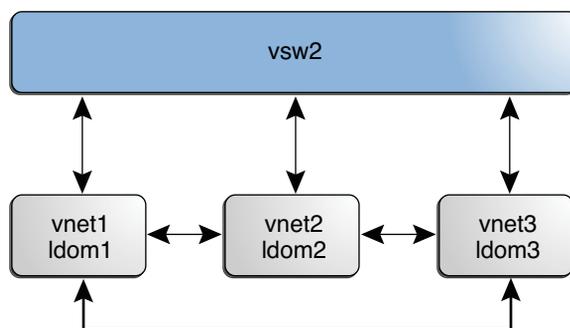
`inter-vnet` チャンネルを割り当てないことで、仮想 I/O デバイスをゲストドメインに追加するために、より多くの LDC チャンネルが利用可能になります。

注- 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。

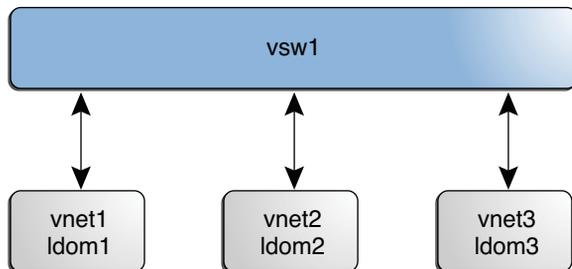
図 8-3 Inter-Vnet チャンネルを使用する仮想スイッチの構成



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。つまり、`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ることがわかります。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 - inter-vnet LDC チャ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 8-4 Inter-Vnet チャンネルを使用しない仮想スイッチの構成



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。

```
# ldm add-vsw [id=switch-id] vswitch-name ldom
# ldm add-vnet [id=network-id] if-name vswitch-name ldom
```

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時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時に `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 nxge0 switch@0 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 - Oracle Solaris 11 システムでは、汎用リンクが netn の形式で vswn 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、vswn 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 OS ネットワークインタフェース名を検索する方法

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。`net-c` に対応する、`ldg1` での Oracle Solaris OS ネットワークインタフェース名を確認するには、次の手順を実行します。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。

- 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。

```
# 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判定するには、`switch@n` の `n` に示された仮想スイッチデバイス番号を探します。

- 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。

```
# uname -n
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 10 OS. `ifconfig` コマンドを使用します。

```
# 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
```

- **Oracle Solaris 11 OS**。dladm コマンドを使用します。

まず、dladm show-phys コマンドを使用して、vnet1 に指定するインタフェースの名前を決定する必要があります。

```
primary# dladm show-phys |grep vnet1
net2          Ethernet          up          0          unknown    vnet1

primary# dladm show-linkprop -p mac-address net2
LINK  PROPERTY  PERM VALUE  DEFAULT  POSSIBLE
net2   mac-address  rw   00:14:4f:f8:dd:68  00:14:4f: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 (vnet)、および仮想スイッチ (vsw) 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。これにより、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。

Logical Domains に割り当てられる MAC アドレスの範囲

Logical Domains 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要求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 - 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。

可能性のあるアドレスを選択する前に、Logical Domains Manager は、自動的に割り当てられ、最近解放されたアドレスが、ここで使用するためにデータベースに保存されているかどうかをまず確認します (165 ページの「解放された MAC アドレス」を参照)。保存されていた場合、Logical Domains Manager はデータベースから候補となるアドレスを選択します。

最近解放されたアドレスが使用できない場合、MAC アドレスはこの用途のために確保された 256K の範囲のアドレス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、164 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、ほかのアドレスの選択および競合の再確認を繰り返し行います。この動作は、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、応答が返されるまで 1 秒待機します。Logical Domains 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。要求を送信した 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を認識し、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、デフォルトの生存期間 (TTL) が 1 である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。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、次の処理を行います。

- 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- 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。

解放された MAC アドレス

自動の MAC アドレスに関連付けられた論理ドメインまたはデバイスが削除されると、その MAC アドレスはそのシステムであとで使用する場合に備えて、最近解放された MAC アドレスのデータベースに保存されます。これらの MAC アドレスを保存して、動的ホスト構成プロトコル (DHCP) サーバーのインターネットプロトコル (IP) アドレスが使い果たされないようにします。DHCP サーバーが IP アドレスを割り当てるとき、しばらくの間 (リース期間中) その動作が行われます。多くの場合、リース期間は非常に長く構成されており、通常は数時間または数日間です。ネットワークデバイスが作成および削除される割合が高く、Logical Domains Manager が自動的に割り当てられた MAC アドレスを再利用しない場合、割り当てられる MAC アドレスの数によって典型的な構成の DHCP サーバーがすぐに圧迫される可能性があります。

Logical Domains Manager は、論理ドメインまたはネットワークデバイスの MAC アドレスを自動的に取得するように要求されると、以前に割り当てられた再利用可能な MAC アドレスが存在するかどうかを確認するために、解放された MAC アドレスデータベースを最初に参照します。このデータベースに使用可能な MAC アドレスが存在する場合、重複した MAC アドレスの検出アルゴリズムが実行されます。以前に解放された MAC アドレスが、そのあと割り当てられていない場合は、その MAC アドレスが再利用され、データベースから削除されます。競合が検出された場合、そのアドレスは単にデータベースから削除されます。Logical Domains Manager は、データベース内の次のアドレスを試行するか、使用可能なアドレスがない場合は、新しい MAC アドレスをランダムに選択します。

Logical Domains 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、167 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 - 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、187 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 準拠かどうかを判別する方法 (Oracle Solaris 10)

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 (vnet) 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。

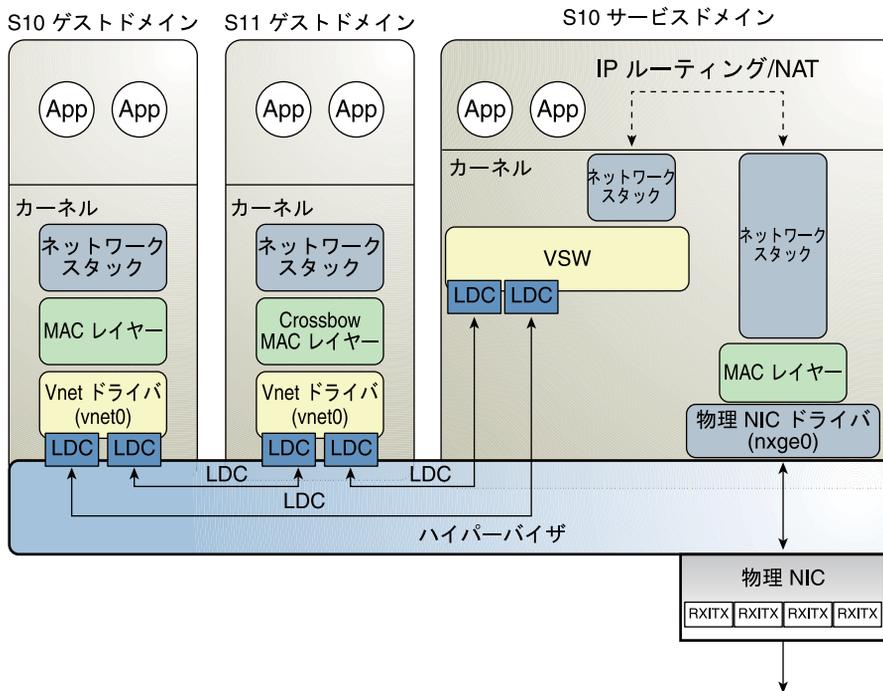
この構成の利点は次のとおりです。

- 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 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8-5 仮想ネットワークルーティング



▼ 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 10)

- 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的なMACアドレスが割り当てられるようにしてください。

```
primary# ldm add-vsw [mac-addr=xx:xx:xx:xx:xx:xx] primary-vsw0 primary
```

- 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、68 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、68 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。

- 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、すべてのドメインに必要なルーティングテーブルを設定します。

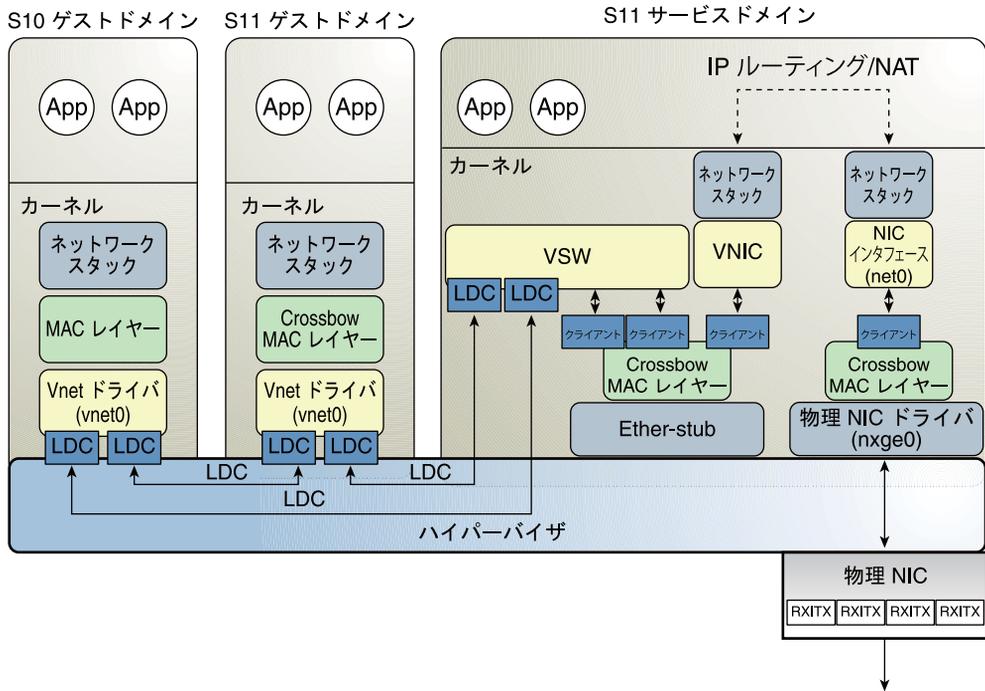
IP ルーティングの詳細については、『[System Administration Guide: IP Services](#)』の「[Packet Forwarding and Routing on IPv4 Networks](#)」を参照してください。

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 とも通信できます。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8-6 仮想ネットワークルーティング



▼ 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。

『Connecting Systems Using Fixed Network Configuration in Oracle Solaris 11.1』の「Setting IP Interface Properties」および『System Administration Guide: IP Services』の「Packet Forwarding and Routing on IPv4 Networks」を参照してください。

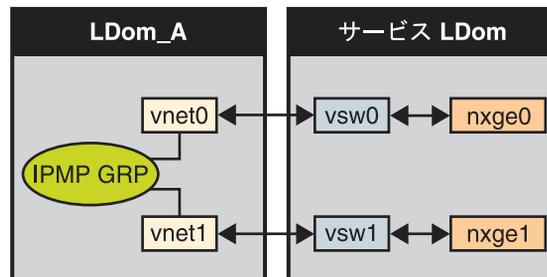
Logical Domains 環境での IPMP の構成

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トコルベースの検出のみを構成できます。

ドメインの IPMP グループへの仮想ネットワークデバイスの構成

次の図は、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、2つの別の物理インタフェースを使用する2つの仮想ネットワーク (vnet0 と vnet1) を示しています。物理インタフェースは、Oracle Solaris 10 では nxge0 および nxge1、Oracle Solaris 11 では net0 および net1 です。図に、Oracle Solaris 10 物理インタフェース名を示します。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。

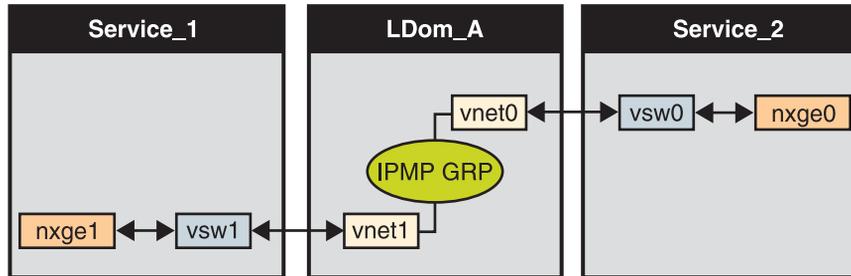
図 8-7 個別の仮想スイッチインスタンスに接続された2つの仮想ネットワーク



注 - 図 8-7 に、Oracle Solaris 10 システムでの構成を示します。Oracle Solaris 11 システムの場合、インタフェース名のみが nxge0 と nxge1 に対してそれぞれ net0 と net1 のように、汎用名を使用するように変更されます。

次の図に示すように、各仮想ネットワークデバイス (vnet0 および vnet1) を異なるサービスドメインの仮想スイッチインスタンスに接続すると、論理ドメインでの信頼性をさらに高めることができます。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 8-8 異なるサービスドメインに接続された各仮想ネットワークデバイス



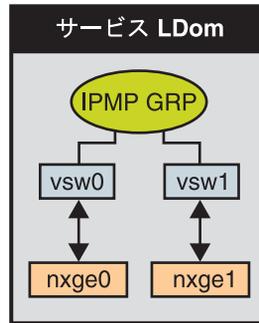
注 - 図 8-8 に、Oracle Solaris 10 システムでの構成を示します。Oracle Solaris 11 システムの場合、インタフェース名のみが nxge0 と nxge1 に対してそれぞれ net0 と net1 のように、汎用名を使用するように変更されます。

詳細は、Oracle Solaris 10 『[System Administration Guide: IP Services](#)』または [Oracle Solaris 11.1 Information Library](#) で Oracle Solaris ネットワークの確立に関する章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

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で、サービスドメインで IPMP を構成できます。次の図に、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 および vsw1) を示します。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。2 つの物理インタフェースは、Oracle Solaris 10 では nxge0 および nxge1、Oracle Solaris 11 では net0 および net1 です。次の図に、Oracle Solaris 10 物理インタフェース名を示します。

図 8-9 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



注 - 図 8-9 に、Oracle Solaris 10 システムでの構成を示します。Oracle Solaris 11 システムの場合、インタフェース名のみが `nxge0` と `nxge1` に対してそれぞれ `net0` と `net1` のように、汎用名を使用するように変更されます。

Logical Domains 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ステータス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ステータス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-1つの Logical Domains 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せん。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。物理リンクステータス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 8/11 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-`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 10 の場合は、『[System Administration Guide: Security Services](#)』の「[Configuring RBAC \(Task Map\)](#)」を参照してください。Oracle Solaris 11.1 の場合は、『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート III 「[Roles, Rights Profiles, and Privileges](#)」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-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。

```
# ldm add-vnet linkprop=phys-state if-name vswitch-name ldom
```

次の例では、論理ドメイン ldom1 の primary-vsw0 に接続された vnet0 で物理リンクステータスの更新を有効にします。

```
# ldm add-vnet linkprop=phys-state 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。

```
# ldm set-vnet linkprop=phys-state if-name ldom
```

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。

```
# ldm set-vnet linkprop=phys-state 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 vnet0 で物理リンクステータスの更新を無効にします。

```
# ldm set-vnet linkprop= vnet0 ldom1
```

例 8-1 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-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。

```
# ldm add-vnet vnet0 primary-vsw0 ldom1
# ldm add-vnet 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 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 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
# 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
```

- 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。

`net0` と `net1` はそれぞれ `vnet0` と `vnet1` の Oracle Solaris 11 バニティー名です。

- **Oracle Solaris 10 OS。** 次のコマンドを使用します。

```
# ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom1
```

- **Oracle Solaris 11 OS。** 次のコマンドを使用します。

```
# ldm add-vnet linkprop=phys-state net0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state net1 primary-vsw1 ldom1
```

注-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
```

Logical Domains 1.3 以前のリリースの IPMP の構成および使用

Logical Domains 1.3 以前のリリースでは、仮想スイッチデバイスおよび仮想ネットワークデバイスはリンク障害の検出を実行できません。それらのリリースでは、プローブベースの IPMP を使用してネットワーク障害の検出と復旧を設定できます。

ゲストドメインでの IPMP の構成

ゲストドメイン内の仮想ネットワークデバイスは、[図 8-7](#) と [図 8-8](#) に示すとおり IPMP グループに構成できます。唯一の相違点は、仮想ネットワークデバイスでテストアドレスを構成することでプローブベースの障害検出が使用されることです。プローブベースの IPMP の構成の詳細は、『[System Administration Guide: 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

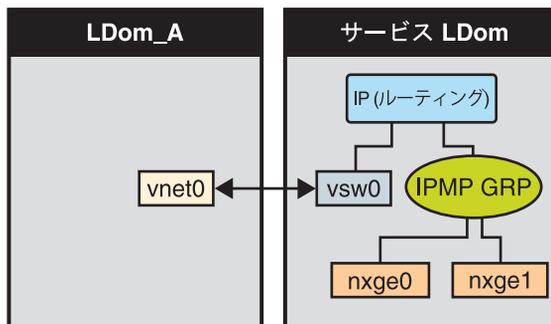
Logical Domains 1.3 以前のリリースでは、仮想スイッチデバイスは物理リンク障害の検出を実行できません。このような場合、サービスドメインの物理インタフェースを IPMP グループに構成することで、ネットワーク障害の検出と復旧を設定できます。これを行うには、物理ネットワークデバイスを割り当てずにサービスドメインの仮想スイッチを構成します。特に、ldm add-vswitch コマンドを使用して仮想スイッチを作成するときに、net-dev (net-dev=) プロパティに値を指定しないでください。サービスドメインに仮想スイッチインタフェースを作成して、サービスドメイン自体が IP ルーターとして機能するように構成します。IP ルーティングの設定については、Oracle Solaris 10 の『[System Administration Guide: IP Services](#)』を参照してください。

いったん仮想スイッチが構成されると、仮想ネットワークから発生し外部のマシンに送信される予定のすべてのパケットは、物理デバイスを使用して直接送信される

のではなく、IP 層に送信されます。物理インタフェースに障害が発生した場合、IP 層は障害を検出し、自動的にセカンダリインタフェースを使用してパケットをふたたび経路指定します。

物理インタフェースは直接 IPMP グループに構成されているため、グループは、リンクベースまたはプローブベースのいずれかの検出用に設定できます。次の図に、IPMP グループの一部として構成された2つのネットワークインタフェース (nxge0 および nxge1) を示します。仮想スイッチインスタンス (vsw0) は、IP 層にパケットを送信するネットワークデバイスとして作成されています。

図 8-10 IPMP グループの一部として構成された2つのネットワークインタフェース



注 - 図 8-10 に、Oracle Solaris 10 システムでの構成を示します。Oracle Solaris 11 システムの場合、インタフェース名のみが nxge0 と nxge1 に対してそれぞれ net0 と net1 のように、汎用名を使用するように変更されます。

▼ プローブベースの IPMP 用のホストルートを構成する方法

注 - この手順は、ゲストドメインおよび 1.3 より前のリリースのみに適用されます。1.3 より前のリリースでは、プローブベースの IPMP のみがサポートされています。

ネットワーク内の IPMP インタフェースに対応するルーターに明示的なルートが構成されていない場合、IPMP プローブベースの検出を目的どおりに動作させるには、ターゲットシステムへの明示的なホストルートを 1 つ以上構成する必要があります。このようにしない場合、プローブ検出がネットワーク障害を検出できないことがあります。

- ホストルートを構成します。

```
# route add -host destination-IP gateway-IP -static
```

たとえば、次のように表示されます。

```
# route add -host 192.168.102.1 192.168.102.1 -static
```

詳細は、『System Administration Guide: IP Services』の「Configuring Target Systems」を参照してください。

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 vnet デバイスには複数の 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。Logical Domains 環境では、この手順のほかに Logical Domains Manager CLI コマンドを使用して、対応する VLAN に vnet を割り当てる必要があります。Logical Domains Manager CLI コマンドの詳細については、[ldm\(1M\)](#) を参照してください。

同様に、サービスドメインの仮想スイッチデバイスに VLAN インタフェースを構成することができます。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。

ゲストドメインに vnet デバイスを作成する場合は、そのデバイスを必要な VLAN に割り当てる必要があります。それには、ldm add-vnet コマンドで pvid= 引数および vid= 引数を使用して、この vnet にポート VLAN ID および 0 個以上の VLAN ID を指定します。これによって、仮想スイッチは、Logical Domains 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。

同様に、vsw デバイス自体が属することになる VLAN をネットワークインタフェースとして作成する場合は、ldm add-vsw コマンドで pvid= 引数および vid= 引数を使用して、vsw デバイス内に構成する必要があります。

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 (PVID)

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、タグなしモードで示します。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vnet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。vnet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vnet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)

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を仮想スイッチおよび仮想ネットワークデバイスに割り当てる方法

- 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。仮想ネットワーク (vnet) を3つのVLANに割り当てます。VLAN 20 をタグなし、VLAN 21 およびVLAN 22 をタグ付きとして構成します。

```
# ldm add-vsw net-dev=nxge0 pvid=21 vid=20 primary-vsw0 primary
# ldm add-vnet pvid=20 vid=21,22 vnet01 primary-vsw0 ldom1
```

- 2 VLAN インタフェースを作成します。

この例では、ドメイン内のこれらのデバイスのインスタンス番号は0で、VLANはこれらのサブネットに対応づけられていることを前提としています。

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)

- a. サービス (primary) ドメインでVLAN インタフェースを作成します。

- Oracle Solaris 10 OS. `ifconfig` コマンドを使用します。

```
primary# ifconfig vsw0 plumb
primary# ifconfig vsw0 192.168.2.100 netmask 0xffffffff broadcast + up
primary# ifconfig vsw20000 plumb
primary# ifconfig vsw20000 192.168.1.100 netmask 0xffffffff broadcast + up
```

- Oracle Solaris 11 OS. `dladm` コマンドと `ipadm` コマンドを使用します。

```
primary# dladm create-vlan -l vsw0 -v20
primary# ipadm create-ip net0
primary# ipadm create-addr -T static -a 192.168.2.100/24 net0/ipv4
primary# ipadm create-ip net20000
primary# ipadm create-addr -T static -a 192.168.1.100/24 net20000/ipv4
```

- b. ゲスト (ldom1) ドメインでVLAN インタフェースを作成します。

- Oracle Solaris 10 OS. `ifconfig` コマンドを使用します。

```
ldom1# ifconfig vnet0 plumb
ldom1# ifconfig vnet0 192.168.1.101 netmask 0xffffffff broadcast + up
ldom1# ifconfig vnet21000 plumb
ldom1# ifconfig vnet21000 192.168.2.101 netmask 0xffffffff broadcast + up
ldom1# ifconfig vnet22000 plumb
ldom1# ifconfig vnet22000 192.168.3.101 netmask 0xffffffff broadcast + up
```

Oracle Solaris 10 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、『[System Administration Guide: IP Services](#)』の「[Administering Virtual Local Area Networks](#)」を参照してください。

- Oracle Solaris 11 OS。dladm コマンドと ipadm コマンドを使用します。

```
ldom1# dladm create-vlan -l net0 -v21
ldom1# ipadm create-ip net0
ldom1# ipadm create-addr -T static -a 192.168.1.101/24 net0/ipv4
ldom1# ipadm create-ip net21000
ldom1# ipadm create-addr -T static -a 192.168.2.101/24 net21000/ipv4
ldom1# ipadm create-ip net22000
ldom1# ipadm create-addr -T static -a 192.168.3.101/24 net22000/ipv4
```

Oracle Solaris 11 OS で VLAN インタフェースを構成する方法の詳細は、『[Managing Oracle Solaris 11.1 Network Performance](#)』の「[Administering VLANs](#)」を参照してください。

▼ 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 に存在する場合に、Oracle Solaris JumpStart 機能を使用して、ネットワークを介して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。インストールサーバーに関連付けられた VLAN ID を、仮想ネットワークデバイスの PVID として指定します。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。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。仮想スイッチは、ネットワークインストールの実行中、ゲストドメインから送受信されるパケットのタグ付きおよびタグなし処理を行います。ネットワークインストールが完了して Oracle Solaris OS がブートしたら、仮想ネットワークデバイスがその VLAN でタグ付けされるように構成できます。その後、その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。

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、78 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 最初に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
```

- 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
```

NIU ハイブリッド I/O の使用

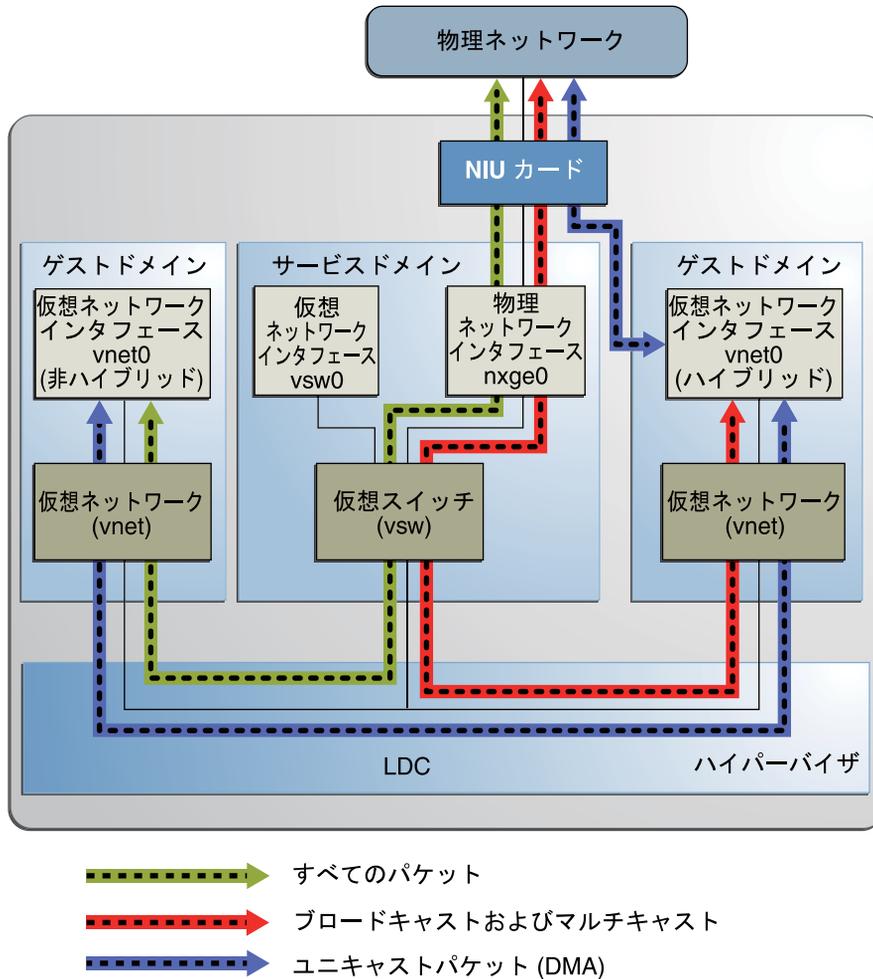
仮想 I/O フレームワークは、機能およびパフォーマンスを向上させるために、「ハイブリッド」I/O モデルを実装しています。ハイブリッド I/O モデルでは、ダイレクト I/O および仮想化 I/O を組み合わせることで、仮想マシンへの柔軟な I/O リソース配備が可能になっています。これは、仮想マシンに対してダイレクト I/O の機能が十分に提供されない場合、または仮想マシンが持続的にあるいは一貫してダイレクト I/O を利用できない場合に特に便利です。この状況は、リソースの可用性または仮想マシンの移行が原因で発生する可能性があります。ハイブリッド I/O アーキテクチャーは、Oracle Sun 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クセス (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 - SR-IOV が推奨されるため NIU ハイブリッド I/O 機能は非推奨となっています。

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 デバイスに貸し出されます。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 - 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 8-11 ハイブリッド仮想ネットワーク接続



注 - 図 8-11 に、Oracle Solaris 10 システムでの構成を示します。Oracle Solaris 11 システムの場合、インタフェース名のみが nxge0 に対して net0 のように、汎用名を使用するように変更されます。

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の注意事項は、次のとおりです。

- vnet デバイスのハイブリッドモードオプションは、提案のみとして扱われます。つまり、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、デバイスがこ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-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。

- デフォルトでは、vnetデバイスのハイブリッドモードは無効になっています。ldmコマンドを使用して、明示的に有効にされる必要があります。[186ページの「ハイブリッドモードを有効にする方法」](#)および[ldm\(1M\)](#)のマニュアルページを参照してください。
- 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。

```
# 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。

```
# 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。

- Oracle Solaris 10 OS。次のコマンドを使用します。

```
# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

- Oracle Solaris 11 OS。次のコマンドを使用します。

次の例では nxge0 の代わりに net2 を使用しています。

```
# ldm add-vsw net-dev=net2 primary-vsw0 primary
```

▼ ハイブリッドモードを有効にする方法

- たとえば、作成中に vnet デバイスのハイブリッドモードを有効にします。

```
# ldm add-vnet mode=hybrid vnet01 primary-vsw0 ldom01
```

▼ ハイブリッドモードを無効にする方法

- たとえば、vnet デバイスのハイブリッドモードを無効にします。

```
# 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の詳細は、『[System Administration Guide: IP Services](#)』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。

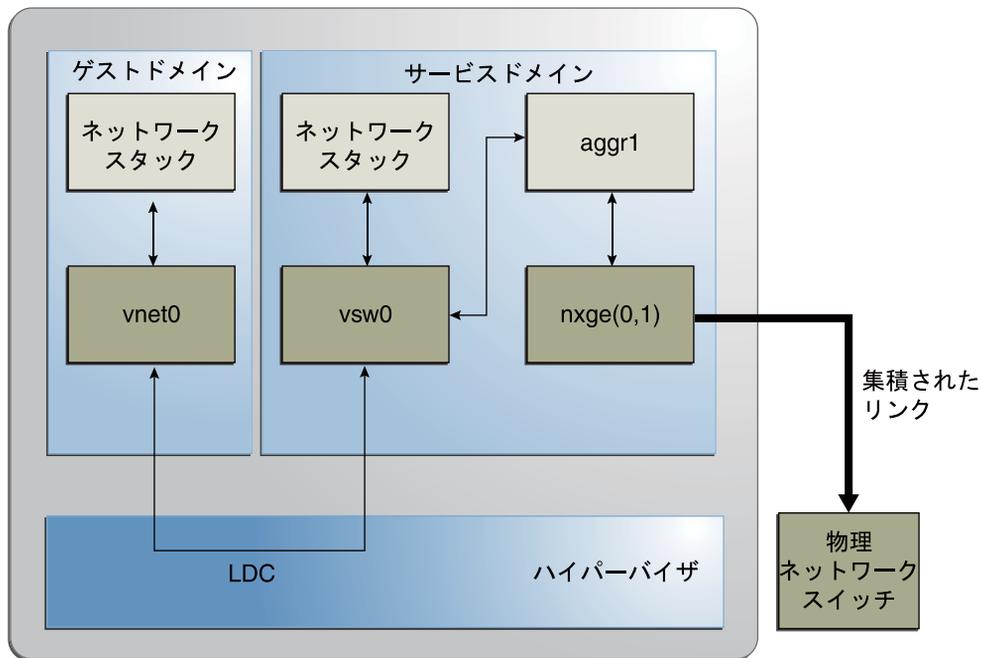
リンクアグリゲーションを仮想スイッチに割り当てると、物理ネットワークに対して送受信されるトラフィックは集積体を通過しています。必要な負荷分散またはフェイルオーバーは、ベースとなる集積体のフレームワークによって透過的に処理されます。リンクアグリゲーションは、ゲストドメイン上の仮想ネットワーク (vnet) デバイスに対して、および集積体を使用する仮想スイッチにバインドされた仮想ネットワークデバイスに対して、完全に透過的です。

注 - 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。68 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。

次の図に、物理インタフェース nxge0 および nxge1 上で集積体 aggr1 を使用するように構成された仮想スイッチを示します。

図 8-12 リンクアグリゲーションを使用する仮想スイッチの構成



注 - 図 8-12 に、Oracle Solaris 10 システムでの構成を示します。Oracle Solaris 11 システムの場合、インタフェース名のみが `nxge0` と `nxge1` に対してそれぞれ `net0` と `net1` のように、汎用名を使用するように変更されます。

ジャンボフレームの構成

Logical Domains 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。この変更によって、これらのドライバのネットワークスループットが向上します。

▼ 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

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。

特定の状況では、仮想ネットワークデバイス上で MTU 値を直接指定できます。これは、仮想ネットワークデバイスで必要な 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に行うことがあります。

注 - Oracle Solaris 10 5/09 OS では、物理デバイスの MTU は仮想スイッチの MTU と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。

- 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 10 の場合は、『[System Administration Guide: Security Services](#)』の「[Configuring RBAC \(Task Map\)](#)」を参照してください。Oracle Solaris 11.1 の場合は、『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート III 「[Roles, Rights Profiles, and Privileges](#)」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、サービスド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。

```
# 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、サービスド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。

```
# 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。

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。

```
# ldm set-vnet mtu= vnet-name ldom
```

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。

例 8-2 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` コマンドによって、仮想スイッチデバイス `primary-vsw0` を MTU 値 9000 で作成します。ネットワークデバイス `nxge 0` のインスタンス 0 は、`net-dev` プロパティの値として指定されています。

```
# ldm add-vsw net-dev=nxge0 mtu=9000 primary-vsw0 primary
```

次に、`ldm add-vnet` コマンドによって、クライアントの仮想ネットワークデバイスをこの仮想スイッチ `primary-vsw0` に追加します。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。

```
# ldm add-vnet vnet01 primary-vsw0 ldom1
```

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 10 OS。** `ifconfig` コマンドによって、サービスドメイン `primary` に仮想スイッチインタフェースが作成されます。`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。

```
# ifconfig vsw0 plumb
# ifconfig vsw0 192.168.1.100/24 up
# 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` コマンドによって、ゲストドメイン `ldom1` に仮想ネットワークインタフェースが作成されます。`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。

```
# ifconfig vnet0 plumb
# ifconfig vnet0 192.168.1.101/24 up
# ifconfig vnet0
vnet0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 4
inet 192.168.1.101 netmask ffffffff broadcast 192.168.1.255
ether 0:14:4f:f9:c4:13
```

- **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` コマンドによって、ゲストドメイン `ldom1` に仮想ネットワークインタフェースが作成されます。`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。

```
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.168.1.101/24 net0/ipv4
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0 mtu ipv4 rw 9000 -- 9000 68-9000
```

- 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 10 OS。** `ifconfig` コマンドを使用します。

```
# ifconfig vnet0 mtu 4000
# 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
```

- **Oracle Solaris 11 OS。** `ipadm` コマンドを使用します。

```
# ipadm set-ifprop -p mtu=4000 net0
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0 mtu ipv4 rw 4000 -- 9000 68-9000
```

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 - 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 - 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ドメインがジャンボフレームをサポートしている Logical Domains 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。これは、クライアントゲストドメインおよびサービ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、Logical Domains 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。次に、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を説明します。

- `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** `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

このデバイスを使用すると、ゲストドメインが Oracle Solaris 11 サービスドメインに構成されているゾーンと通信できます。

- 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 `vswn` デバイスと `vnetn` デバイスに汎用名を割り当てます。そのため、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

`vswn` デバイスと `vnetn` デバイスでは、VNIC を使用できません。`vswn` および `vnetn` に VNIC を作成しようとすると失敗します。『Oracle VM Server for SPARC 3.0 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。

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。

- 196 ページの「ドメインの移行の概要」
- 196 ページの「移行処理の概要」
- 197 ページの「ソフトウェアの互換性」
- 197 ページの「移行処理のセキュリティー」
- 198 ページの「ドメインの移行」
- 199 ページの「アクティブなドメインの移行」
- 204 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」
- 198 ページの「予行演習の実行」
- 205 ページの「進行中の移行のモニタリング」
- 206 ページの「進行中の移行の取り消し」
- 206 ページの「移行の失敗からの回復」
- 198 ページの「対話型でない移行の実行」
- 207 ページの「移行の例」

注 - 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 3.0 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、コ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

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。

- バージョン 2.1 以上の Logical Domains Manager が両方のマシンで実行されている必要があります。
- 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。『Oracle VM Server for SPARC 3.0 リリースノート』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。移行操作を 2 台のマシンで実行するため、ソースマシンとターゲットマシンの両方でユーザーの認証が必要です。特に、スーパーユーザー以外のユーザーは LDom Management 権利プロファイルを使用する必要があります。

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 9-2 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。198 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。

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。

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、[199 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、[204 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。

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- 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する可能性があります。

対話型でない移行の実行

`ldm migrate-domain -p filename` コマンドを使用して、対話型でない移行操作を開始できます。

`-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、『[Oracle VM Server for SPARC 3.0 リリースノート](#)』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。各プライマリドメインのCPUを16個以上にするのが最善ですが、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。

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。
- Oracle Solaris 10 OS を実行するシステムの場合、ソースマシンとターゲットマシンでプロセッサタイプが同じである必要があります。
- Oracle Solaris 11 OS の場合、`cpu-arch` プロパティを設定すると、プロセッサタイプが異なるシステム間で移行できます。次の `cpu-arch` プロパティ値がサポートされています。
 - `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。
 - `generic` は一般的な 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。

`generic` 値は、`native` 値と比較して、パフォーマンスが低下することがあります。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せん。

プロセッサタイプを確認するには、次のように `psrinfo -pv` コマンドを使用します。

```
# psrinfo -pv
The physical processor has 8 virtual processors (0-7)
  SPARC-T4 (chipid 0, clock 2548 MHz)
```

- 移行するドメインで Oracle Solaris 11 OS を実行している場合、プロセッサ周波数と STICK 周波数値が異なるソースシステムとターゲットシステム間でドメインを移行できます。cpu-arch プロパティ値が設定されていなくてもこのタイプの移行は可能です。ただし、移行するドメインで Oracle Solaris 10 OS を実行している場合は、プロセッサ周波数と STICK 周波数値が一致している必要があります。

STICK 周波数を確認するには、次のように `prtconf -pv` コマンドを使用します。

```
# prtconf -pv | grep stick-frequency
stick-frequency: 05f4bc08
```

注 - STICK レジスタがインクリメントする周波数は、フルスピードの CPU 周波数から導かれます。ただし、両方のマシンで CPU 周波数が同じであっても、正確な STICK レジスタ周波数が若干異なるために移行がブロックされる可能性があります。

メモリーの移行要件

ターゲットマシン上に、ドメインの移行に対応できる十分な空きメモリーが存在する必要があります。さらに、移行が終了するまで次に示すいくつかのプロパティが維持される必要があります。

- 同じ数、同じサイズのメモリーブロックを作成する必要があります。
- メモリーブロックの物理アドレスが一致する必要はありませんが、移行が終了するまで同じ実アドレスが維持される必要があります。

また、ターゲットマシンの使用可能メモリーのレイアウトと移行元ドメインのメモリーのレイアウトに互換性がある必要があります。互換性がないと、移行は失敗します。特に、ターゲットマシンのメモリーが複数の小さいアドレス範囲に分割されているのに、移行元ドメインには単一の大きいアドレス範囲が必要な場合、移行は失敗します。次の例は、この場合について示したものです。ターゲットマシンの 2 つのメモリーブロックに、2G バイトの空きメモリーがあるとします。

```
# ldm list-devices memory
MEMORY
```

PA	SIZE
0x108000000	1G
0x188000000	1G

移行元ドメイン ldg-src にも 2G バイトの空きメモリーがありますが、これは単一のメモリーブロックに配置されています。

```
# ldm list -o memory ldg-src
NAME
ldg-src

MEMORY
  RA          PA          SIZE
  0x8000000  0x208000000  2G
```

このようなメモリーレイアウトの場合、移行は失敗します。

```
# ldm migrate-domain ldg-src t5440-sys-2
Target Password:
Unable to bind 2G memory region at real address 0x8000000
Domain Migration of LDom ldg-src failed
```

注 - Oracle Solaris 11.1 以降では、移行されたドメインでメモリーの動的再構成 (DR) を有効にするためにリポートする必要はありません。Oracle Solaris 11.1 よりも前では、リポートするまで、移行されたドメインのメモリーの DR は無効になっています。リポートが完了すると、移行されたドメインでメモリー DR が再度有効になります。

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。

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。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-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りポートシステムに到達できるように、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、制御ドメインの最初の 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。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。

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 が推奨されるため NIU ハイブリッド I/O 機能は非推奨になりました。

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。

次の Oracle Solaris OS バージョンでは、暗号化装置の DR がサポートされています。

- Solaris 10 10/09 以降の OS
- Solaris 10 5/08 OS とパッチ ID 142245-01 以降

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-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、遅延再構成処理はブロックされます。

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

Oracle VM Server for SPARC 3.0 以降のリリースでは、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。

以前の Oracle VM Server for SPARC リリースでは、PM のエラスティックポリシーが有効なソースマシンまたはターゲットマシンでは、ドメインの移行がサポートされていません。移行の処理中にソースマシンまたはターゲットマシンの PM ポリシーがパフォーマンスからエラスティックに切り替えられた場合、ポリシーの切り替えは

移行が完了するまで延期されます。ソースマシンまたはターゲットマシンでエラスティックポリシーが有効な間に、ドメインの移行が試みられた場合、移行コマンドはエラーを返します。

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ガ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 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ガ (kldb) で実行中の場合、この調整は不可能です。そのため、移行元ドメインの CPU が 1 つだけではない場合、移行の試行は失敗します。移行元ドメインの CPU が 1 つだけの場合、特定の要件および制限が満たされているときは移行が継続されます。『Oracle VM Server for SPARC 3.0 リリースノート』の「ドメイン移行の制限」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。

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、メモリー、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。

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。

直接 I/O (Direct I/O、DIO) 機能については、89 ページの「PCIe エンドポイントデバイスの割り当て」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。

- 移行元ドメインの場合は、移行のソースであることを示す **s** が表示されます。
- 移行先ドメインの場合は、移行のターゲットであることを示す **t** が表示されます。
- ユーザーによる介入を必要とするエラーが発生した場合、**e** が表示されます。

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%         t5440-sys-2
```

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。

注- 移行が開始された後は、ldm プロセスを中断しても、処理は一時停止しません。これは、移行に影響を与えるのは、ソースマシンおよびターゲットマシン上のデーモン (ldmd) であり、ldm プロセスではないためです。ldm プロセスは、戻る前に、移行が完了したことを示す ldmd からの信号を待機します。

移行の失敗からの回復

ネットワーク接続が次の状況で失われた場合、移行処理は終了します。

- 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
- ただし、ドメインが再開されたことを移行先ドメインが確認できる前

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9-1 ゲストドメインの移行

この例では、`ldg1` ドメインを `t5440-sys-2` という名前のマシンに移行する方法を示します。

```
# ldm migrate-domain ldg1 t5440-sys-2
```

```
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 t5440-sys-2
```

`-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` はターゲットマシン `t5440-sys-2` のパスワードを格納しています。

例9-2 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、ソースマシンの `ldg-src` ドメインの名前を、ターゲットマシン (`t5440-sys-2`) では `ldg-tgt` に変更します。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@t5440-sys-2:ldg-tgt
```

```
Target Password:
```

例9-3 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
```

例9-3 移行の失敗メッセージ (続き)

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9-4 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 t5440-sys-1 です。

```
# ldm list -o status ldg-tgt
NAME
ldg-tgt

STATUS
  OPERATION    PROGRESS    SOURCE
  migration    55%         t5440-sys-1
```

例9-5 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 t5440-sys-2 です。

```
# ldm list -o status -p ldg-src
VERSION 1.6
DOMAIN|name=ldg-src|
STATUS
|op=migration|progress=42|error=no|target=t5440-sys-2
```

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。

- 209 ページの「リソースの再構成」
- 211 ページの「リソースの割り当て」
- 211 ページの「CPU の割り当て」
- 215 ページの「ハードパーティションによるシステムの構成」
- 222 ページの「ドメインへの物理リソースの割り当て」
- 226 ページの「メモリーの動的再構成の使用」
- 234 ページの「電源管理の使用」
- 234 ページの「動的なリソース管理の使用」
- 238 ページの「ドメインリソースの一覧表示」

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 (Operating System、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 8/11 OS および Oracle Solaris 11 OS 以上でサポート
- メモリー - Oracle VM Server for SPARC 2.0 以降のリリースでサポート (226 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 2.0 以降のリリースでサポート (『Oracle VM Server for SPARC 3.0 リリースノート』の「必須および推奨される Oracle Solaris OS バージョン」を参照)
- 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

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限り、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 - 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 3.0 リリースノート』の「制御ドメインから多数の CPU を削除すると失敗する」を参照してください。制御ドメインから多くのメモリーを削除するには、228 ページの「制御ドメインのメモリーを減らす」を参照してください。

リソースの割り当て

Oracle VM Server for SPARC 2.0以降のリリースでは、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るというリソース割り当てメカニズムを採用しています。

「リソースの割り当ての制約」は厳格な要件で、ドメインにリソースを割り当てるときに、システムがこの要件を必ず満たしている必要があります。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。

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だけの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されていることを確認することをお勧めします。

- 1 ドメインにコア全体の制約を適用します。

```
# ldm set-core 1 domain
```

- 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。

```
# ldm ls -o resmgt domain
```

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 10-1 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。

```
# ldm set-core 1 ldg1
# ldm ls -o resmgt ldg1
NAME
ldg1

CONSTRAINT
cpu=whole-core
max-cores=unlimited
threading=max-throughput
```

▼ 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されていることを確認することをお勧めします。

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。

```
# ldm set-domain max-cores=max-number-of-CPU-cores domain
```

注-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。

```
# ldm ls -o resmgt domain
```

- 3 ドメインをバインドし、再起動します。

```
# ldm bind domain
# ldm start domain
```

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 10-2 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。

```
# ldm set-domain max-cores=3 ldg1
# ldm ls -o resmgt ldg1
NAME
ldg1

CONSTRAINT
  cpu=whole-core
  max-cores=3
  threading=max-throughput
```

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。

```
# ldm stop ldg1
# ldm unbind ldg1
# 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。

```
# 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。

```
# ldm bind ldg1
# 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。

- 214 ページの「CPUの動的再構成」
- 214 ページの「動的なリソース管理」

- 214 ページの「ドメインの移行」
- 215 ページの「電源管理」

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、動的なリソース管理(DRM)との互換性はありません。コア全体の制約を使用するドメインでDRMポリシーが有効化された場合、そのポリシーは自動的に無効になります。コア全体の制約は有効なままです。

コア全体の制約が有効なときにはDRMポリシーを有効にできませんが、ドメインのDRMポリシーを定義することはできます。ポリシーは自動的に無効になっても、アクティブのままです。コア全体の制約なしにドメインが再起動されると、このポリシーは自動的に再度有効になります。

次に示すのは、コア全体の制約とDRMの間の予期される相互作用です。

- ドメインにコア全体の制約が設定されている場合、そのドメインでDRMポリシーの有効化を試みると警告メッセージが表示されます。
- アクティブでないドメインでDRMポリシーが有効になっている場合、そのドメインでコア全体の制約を有効にできます。ドメインがアクティブになったときにポリシーが有効だと、システムはそのドメインのDRMポリシーを自動的に無効にします。
- アクティブなドメインまたはバインドされているドメインでDRMポリシーが有効になっている場合、コア全体の制約を有効にすることはできません。

ドメインの移行

コアの全体で構成されているドメインを移行する場合、移行の完了後に、ターゲットドメインでハードパーティション分割を使用するように再構成する必要があります。さらに、ライセンス契約で、ソースシステムとターゲットシステムの両方でドメインの使用が許可されていることを確認する必要があります。

電源管理

コア全体の制約には、電源管理 (PM) のパフォーマンスおよびエラスティックポリシーとの完全な互換性があります。エラスティックポリシーが有効なとき、PM サブシステムは、コア全体の制約で構成されているドメインに対する CPU コアの追加または削除を行うことができます。この場合、コア全体の制約は引き続き適用され、この制約を使用するドメインはコア全体でのみ構成されます。

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。** SPARC T シリーズおよび Fujitsu M10 システムで Oracle VM Server for SPARC ソフトウェアが実行されます。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。
 - ドメインで 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場合、ドメインを CPU コア全体で構成する必要があります。
 - ドメインで 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場合、そのドメインは 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

このセクションのタスクでは、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方法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説明します。

▼ 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方法

- 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します。

```
# ldm list -o resgmt domain
```

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。

例 10-3 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

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。

```
# ldm list -o resgmt ldg1
NAME
ldg1

CONSTRAINT
  whole-core
  max-cores=5
```

▼ ドメインに割り当てられているCPUコアを表示する方法

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コアを表示します。

```
# ldm list -o core domain
```

例 10-4 ドメインに割り当てられているCPUコアの表示

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。

```
# 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。

注 - コア全体を割り当てるために使用する `ldm` サブコマンドが Oracle VM Server for SPARC 2.2 リリースで変更されました。

このセクションのタスクと例では、Oracle VM Server for SPARC 2.2 ソフトウェアで導入された新しいコマンドを使用します。

Logical Domains Manager のバージョン 2.0 または 2.1 を使用して、コア全体をドメインに割り当てる場合、`ldm add-core`、`ldm set-core`、`ldm remove-core` コマンドの代わりに、`ldm add-vcpu -c`、`ldm set-vcpu -c`、`ldm remove-vcpu -c` コマンドをそれぞれ使用します。

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。

Oracle VM Server for SPARC 2.2 リリースから、CPU キャップと CPU コアの割り当てが個別のコマンドで処理されるようになりました。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、コアに割り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は、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- オプションでコアの最大数の制約を設定する場合に必要なことは、ドメインの停止およびバインド解除だけです。

- 1 ドメインを作成します。
`# ldm create domain`
- 2 ドメインの CPU 全体のコアの数を設定します。
`# 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。
`# 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。
`# ldm bind domain`
`# ldm start domain`

例 10-5 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、[218 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。

```
# ldm create ldg1
# ldm set-core 2 ldg1
...
# ldm bind ldg1
# 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、CPU コア全体を使用するようにその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。

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

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。
- 2 ドメインのCPU全体のコアの数を設定します。

```
# 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。

```
# 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。

```
# ldm bind domain
# ldm start domain
```

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。

例10-6 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。

```
# ldm set-core 4 ldg1
```

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。

▼ CPUコア全体で **primary** ドメインを構成する方法

`primary` ドメインがCPUスレッドを使用するように構成されている場合、CPUコア全体を使用するようにその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。

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

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。
- 2 **primary** ドメインのCPUコア全体の数を設定します。

```
# 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。

```
# 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。

適切な手順を使用して、システム構成に依存する primary ドメインをリブートします。93 ページの「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 10-7 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 目目のコマンドは max-cores プロパティを 2 に設定し、4 目目のコマンドは primary ドメインをリブートします。

```
# ldm start-reconf primary
# ldm set-core 2 primary
# ldm set-domain max-cores=2 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

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、個々の CPU スレッドではなく、CPU コア全体のみを追加または削除できます。そのため、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持されます。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 の最大数を超えようとする、失敗します。

注-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限り、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、設定、または削除するには、次のコマンドを使用します。

```

ldm add-core number-of-cpu-cores domain
ldm set-core number-of-cpu-cores domain
ldm rm-core number-of-cpu-cores domain

```

注-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 10-8 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。

```

# ldm list ldg1
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1      active -n---- 5000  16    2G      0.4%  5d 17h 49m
# ldm list -o resmgmt ldg1
NAME
ldg1

CONSTRAINT
  whole-core
    max-cores=4
# 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)
# ldm add-core 2 ldg1
# 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。DRMを使用する場合、DRM ポリシーは、CPUコア全体で構成されたドメインに適用されません。

DRM ポリシーに CPU コア全体で構成されたドメインを含めることができます。ただし、そのようなポリシーをアクティブにしても、そのドメインに対し、自動的に無効にされます。あとでドメインを CPU コア全体ではなく、CPU スレッドで構成しない限り、ドメインは CPU コア全体で構成されたままになります。ドメインを CPU スレッドを使用するように構成すると、DRM ポリシーがそのドメインに対して自動的に再有効にされます。

CPU 電源管理

CPU コア全体で構成されたドメインに対し、パフォーマンスまたはエラスティックポリシーを有効にして、CPU 電源管理 (PM) を使用できます。PM を使用すると、システムのハードパーティション分割が維持されます。

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。

ドメイン移行の非互換性

コアの全体で構成されているドメインを移行する場合、移行の完了後に、ターゲットドメインでハードパーティション分割を使用するように再構成する必要があります。さらに、ライセンス契約で、ソースシステムとターゲットシステムの両方でドメインの使用が許可されていることを確認する必要があります。

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 3.0 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。225 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、226 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ーに関して十分な知識のある管理者のみが使用してください。この高度な構成機能は、特定の割り当て規則を適用し、システムのパフォーマンス全体に影響す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[,...]] ldom
# ldm set-core cid=core-ID[,core-ID[,...]] ldom
# ldm rm-core [-f] cid=core-ID[,core-ID[,...]] ldom
```

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。

- 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ーに関して十分な知識のある管理者のみが使用してください。この高度な構成機能は、特定の割り当て規則を適用し、システムのパフォーマンス全体に影響す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[,...]] ldom
# ldm set-mem mblock=PA-start:size[,PA-start:size[,...]] ldom
# ldm rm-mem mblock=PA-start:size[,PA-start:size[,...]] ldom
```

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 - 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、[225 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。

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。mblock プロパティを設定して、physical-bindings 制約を memory に設定します。同様に、cid プロパティを設定して、physical-bindings 制約を core に設定します。cid プロパティと mblock プロパティの両方を設定した場合、physical-bindings 制約が core, memory に設定されます。

▼ 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。

```
# ldm unbind domain
```
- 2 名前付きリソースを削除します。
 - 名前付きコア。

```
# ldm set-core cid=core-ID domain
```
 - 名前付きメモリー。

```
# ldm set-mem mblock=PA-start:size domain
```
- 3 CPU またはメモリーリソースを追加します。
 - CPU リソース。

```
# ldm add-vcpu number domain
```
 - メモリーリソース。

```
# ldm add-mem size[unit] domain
```
- 4 ドメインを再バインドします。

```
# ldm bind domain
```

▼ 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。

```
# ldm unbind domain
```

- 2 リソースの数を 0 に設定します。

- CPU リソース。

```
# ldm set-core 0 domain
```

- メモリーリソース。

```
# ldm set-mem 0 domain
```

- 3 物理的にバインドされている CPU またはメモリーを追加します。

- CPU リソース。

```
# ldm add-core cid=core-ID domain
```

- メモリーリソース。

```
# ldm add-mem mblock=PA-start:size domain
```

- 4 ドメインを再バインドします。

```
# ldm bind domain
```

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)のリポートを実行する必要があります。

注 - 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、無限のメモリー割り当てを実行できません。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は行なわれます。
- 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。
- `ldm add-core` コマンドまたは `ldm set-core` コマンドを使用することでしか、物理リソースをドメインに割り当てることができません。
- `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

Oracle VM Server for SPARC 2.0 リリースには、メモリーの動的再構成が導入されています。この機能は容量に基づき、アクティブな論理ドメインに対する任意の量のメモリーの追加または削除を可能にします。

次に示すのは、メモリー DR 機能を使用するための要件と制限です。

- 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。229 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。230 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。

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。操作の進捗状況を追跡し、実行中のメモリー DR 要求を取り消すこともできます。

メモリー DR 要求の進捗状況の追跡

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。

メモリー DR 要求の取り消し

`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-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ーの再構成

メモリー DR 機能を使用して、制御ドメインのメモリーを再構成することができます。制御ドメインでメモリー 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。

メモリー 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ーを削除する場合は、メモリー 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ーを減らすようにしてください。

制御ドメインのメモリーを減らす

制御ドメインのメモリーを出荷時デフォルト構成から減らすには、メモリー 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ー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、要求されたメモリー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。

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ー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、メモリーDRをサポートしないドメインについてのメモリー再構成要求はすべて拒否されます。メモリーDRをサポートしない制御ドメインでのメモリー再構成要求は、遅延再構成要求に変換されます。

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。つまり、操作の結果、ドメインのメモリーサイズが指定よりも大きくなる場合があります。

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。つまり、操作の結果、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。

アクティブでないドメインのメモリー配置

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、結果のメモリーサイズが 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。229 ページの「[バインドされているドメインのメモリー配置](#)」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ーの追加

メモリー 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。このため、この単位に調整されていないメモリーを、メモリー 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ーがこの単位に調整されていない場合、それらをメモリー DR を使用してアクティブなドメインに追加することはできません。

単位が調整されたすべてのメモリー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ー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ー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、残りの 2 つのメモリーブロックを追加します。最初のコマンドは、制御ドメインで遅延再構成操作を開始します。2 番目のコマンドは、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。5 番目のコマンドは、もう 1 つの 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。

```
# 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.
```

```
# 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.
-----
```

```
# ldm list
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active    -ndcv- SP     8     2688M  0.1%  23d 8h 8m

# 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

# ldm add-mem 128M ldom1
# 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10-9 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。最初の `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。

`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。

```
# 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

# ldm add-mem 200M ldom1
The size of memory must be a multiple of 256MB.

# 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.

# ldm list
```

例 10-9 アクティブなドメインでのメモリー DR 操作 (続き)

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 --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.
```

```
# 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 10-10 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 256M バイトの境界まで切り下げられます。

```
# 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		

```
# ldm add-mem 100M ldom2
```

```
# 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		

```
# 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.
```

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	4	27392M	0.4%	1d 22h 55m

例 10-10 バインドされたドメインでのメモリー DR 操作 (続き)

```
ldom1          active   -n---- 5000    2    2G      0.5%  1d 1h 25m
ldom2          bound    ------ 5001    2    512M

# ldm rm-mem 100M ldom2
# 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

# 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.

# 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 10-11 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。最初の `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。

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。

```
# 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

# 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.

# ldm set-mem --auto-adj 3400M primary
Adjusting request size to 3.4G.
```

例 10-11 ドメインのメモリーサイズの設定 (続き)

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.

```
# ldm set-mem 690M ldom2
```

```
# 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		

```
# 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.

```
# 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		

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。

PM 機能および ILOM 機能の詳細は、次を参照してください。

- [第 15 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。
- コア全体の制約が構成されているドメインには、DRM ポリシーは適用されません。whole-core 制約が設定されているドメインで DRM の使用を試みると、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しか DRM を使用できません (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 ldom...
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 ldom...
ldm remove-policy [name=policy-name... ldom
```

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。

- 空き CPU リソースがプールにあります。この状況では、単一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、priority プロパティによって、有効にする DRM ポリシーを決定します。
- 空き CPU リソースがプールにありません。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せん。

- 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 10-12 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。使用率がもっとも高いのは、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。

例 10-12 リソース管理ポリシーの追加 (続き)

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。

```
# 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。

例 10-12 リソース管理ポリシーの追加 (続き)

- `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。

```
# 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 ldm1
```

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、および実際と同様の出力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。詳細は、[241 ページ](#)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方法 (-p)」を参照してください。

▼ `ldm` サブコマンドの構文の使用法を表示する方法

- `ldm` のすべてのサブコマンドの構文の使用法を確認します。

```
# 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。次に、左から順に6つの列のそれぞれに表示される可能性のある値を示します。

列1 - ドメインの起動または停止

- `s` 起動または停止
- `-` 可変部分

列2 - ドメインのステータス

- `n` 通常
- `t` 切り替え

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
- `-` 可変部分

列4 - 制御ドメイン

- `c` 制御ドメイン
- `-` 可変部分

列5 - サービスドメイン

- `v` 仮想 I/O サービスドメイン
- `-` 可変部分

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
- `-` 可変部分

利用統計情報の定義

`ldm list` コマンドの長形式 (`-l`) オプションでは、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 によってサイクルのスキップが 4/8 に調整されると、100% の利用率が 50% の利用率と等しくなります。つまり、結果的に CPU のサイクルは利用可能な数の半分になります。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

▼ ソフトウェアのバージョンを表示する方法 (-v)

- 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します。
`ldm -v`

▼ 省略形式のリストを生成する方法

- すべてのドメインの省略形式のリストを生成します。
`ldm list`

▼ 長形式のリストを生成する方法 (-l)

- すべてのドメインの長形式のリストを生成します。
`ldm list -l`

▼ 拡張リストを生成する方法 (-e)

- すべてのドメインの拡張リストを生成します。
`ldm list -e`

▼ 解析可能でマシンが読み取り可能なリストを生成する方法 (-p)

- 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します。

```
# ldm list -p
```

▼ 長形式のリストのサブセットを生成する方法 (-o format)

- 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します。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。

```
# ldm list -o resource[,resource...] ldom
```

- 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 - 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。
- resmgmt - 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 (vdpccl)、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 情報のリスト

- ```
ldm list -o cpu primary
```
- ゲストドメインのドメイン情報のリスト
- ```
# ldm list -o domain ldm2
```
- ゲストドメインのメモリーおよびネットワーク情報のリスト
- ```
ldm list -o network,memory ldm1
```
- ゲストドメインの DRM ポリシー情報のリスト
- ```
# ldm list -o resmgmt,stats ldm1
```

▼ 変数を一覧表示する方法

- ドメインの変数とその値を表示します。

```
# ldm list-variable variable-name ldom
```

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。

```
# ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

▼ バインドを一覧表示する方法

- ドメインにバインドされたリソースを一覧表示します。

```
# ldm list-bindings ldom
```

▼ 構成を一覧表示する方法

- SP に格納されている論理ドメイン構成を一覧表示します。

例 10-13 構成リスト

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。

構成の詳細は、248 ページの「[Logical Domains 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。

```
# ldm list-config
factory-default
3guests
foo [next poweron]
primary
reconfig-primary
```

参考 ラベルの意味

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。

▼ デバイスを一覧表示する方法

- 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)を一覧表示します。

```
# ldm list-devices -a
```

▼ 使用可能なメモリーを一覧表示する方法

- 割り当て可能なメモリーの量を一覧表示します。

```
# ldm list-devices mem
MEMORY
  PA                SIZE
  0x14e000000      2848M
```

▼ サービスを一覧表示する方法

- 使用可能なサービスを一覧表示します。

```
# 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り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方法

- 1つのドメインの制約を一覧表示します。

```
# ldm list-constraints ldom
```

▼ 制約を **XML** 形式で一覧表示する方法

- 特定のドメインの制約を **XML** 形式で一覧表示します。

```
# ldm list-constraints -x ldom
```

▼ 制約をマシンが読み取り可能な形式で一覧表示する方法

- すべてのドメインの制約を解析可能な形式で一覧表示します。

```
# ldm list-constraints -p
```

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。

- 245 ページの「将来の再構築用のドメイン構成の保存」
- 248 ページの「Logical Domains 構成の管理」

将来の再構築用のドメイン構成の保存

基本的な処理は、各ドメインのリソース制約情報を XML ファイルに保存することです。たとえば、ハードウェアの障害のあとに、この XML ファイルを Logical Domains Manager に対して再実行して、必要な構成を再構築できます。

246 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`)」は、制御 (`primary`) ドメインではなく、ゲストドメインに対して有効です。`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せんが、それを `ldm add-domain i` コマンドに指定することはできません。ただし、`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` ドメインを再構成できます。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他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。次の手順を完了すると、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。

▼ ドメイン構成を保存する方法

次の手順は、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方法を示しています。

- 1つまたは複数のドメインのドメイン構成の保存
 - 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 ldom >ldom.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
```

▼ XMLファイルからのドメイン構成の復元方法 (ldm add-domain)

この手順の代わりに、ldm init-system コマンドを使用して、XMLファイルからドメイン構成を復元できます。247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 ldom.xml
```
- 2 ドメインをバインドします。

```
# ldm bind-domain [-fq] ldom
```

-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 ldom
```

例 11-1 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 ファイルと ldm init-system コマンドを使用して、以前に保存した構成を再作成する方法を示します。XML ファイルは、1つまたは複数のドメインの構成を示します。ldm ls-constraints -x コマンドを実行して、XML ファイルを作成できます。ldm init-system コマンドは、factory-default 構成での実行を想定していますが、XML ファイルからあらゆる構成を復元できます。ファイルで指定されたとおりに、primary ドメインが再構成されます。XML ファイルで構成されている primary 以外のドメインは再構成されますが、無効のままです。

この手順の代わりに、ldm add-domain コマンドを使用して、XML ファイルから1つのドメイン構成を復元できます。246 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)」を参照してください。

- 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、45 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 10 の場合は、『System Administration Guide: Security Services』の「Configuring RBAC (Task Map)」を参照してください。Oracle Solaris 11.1 の場合は、『Oracle Solaris 11.1 Administration: Security Services』のパート III 「Roles, Rights Profiles, and Privileges」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 ドメインをリポートする必要があります。-r オプションは、構成後に primary ドメインをリポートします。-r オプションを指定しない場合は、手動でリポ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。

例 11-2 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。他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
```

システムのリポート後、次のコマンドは、ldg1 および ldg2 ドメインをバインドしてリポートします。

```
# ldm bind ldg1
# ldm start ldg1
# ldm bind ldg2
# ldm start ldg2
```

Logical Domains 構成の管理

Logical Domains 「構成」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、ドメインのリポートや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。

Oracle VM Server for SPARC 3.0 以降のリリースでは、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読み込むために使用されます。

注 - 制御ドメインのファイルシステムでは、ミラー化や RAID などのテクノロジーを使用して、ファイルシステムが単一障害点にならないようにしてください。

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。特定の構成をブートすることで、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。

Logical Domains 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
- 実際の構成の変更が、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、次回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 - 電源管理、FMA、ASR、および PRI 更新イベントでは、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。

注 - 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、[259 ページ](#)の「サービスプロセッサでの Logical Domains の使用」を参照してください。

▼ 自動回復ポリシーを変更する方法

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。`autorecovery_policy` プロパティには次の値を使用できます。

- `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。ユーザーは、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。ユーザーは、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。この処理は、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。

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 10 の場合は、[『System Administration Guide: Security Services』](#) の「[Configuring RBAC \(Task Map\)](#)」を参照してください。Oracle Solaris 11.1 の場合は、[『Oracle Solaris 11.1 Administration: Security Services』](#) のパート III 「[Roles, Rights Profiles, and Privileges](#)」を参照してください。

3 `autorecovery_policy` プロパティ値を表示します。

```
# 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 11-3 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
```


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に関する情報とタスク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。

- 253 ページの「CLIでの名前を入力」
- 254 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」
- 255 ページの「コンソールグループの使用」
- 256 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」
- 257 ページの「Oracle VM Server for SPARC による Oracle Solaris OS の運用」
- 259 ページの「サービスプロセッサでの Logical Domains の使用」
- 260 ページの「ドメインの依存関係の構成」
- 264 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」
- 266 ページの「ユニバーサル固有識別子の使用」
- 267 ページの「仮想ドメイン情報コマンドと API」

CLIでの名前を入力

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 (*ldom*)、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。

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 host-name 5001
```

注- 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- 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的な 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 - 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。vntsd を管理するための SMF の構成と使用法、およびコンソールグループの使用法については、Oracle Solaris 10 OS の 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
```

注 - Oracle VM Server for SPARC 3.0 以降のリリースでは、`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、ドメインを停止する必要があります。Oracle Solaris OS の `halt` コマンドを使用して、ドメインを停止できます。

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` に保存する方法

- 次のコマンドを使用します。

```
# ldm add-config config-name
```

電源管理されているドメインのアクティブな CPU での `psradm(1M)` コマンドの使用禁止

`psradm` コマンドを使用して、電源管理されたドメインのアクティブな CPU の動作ステータスを変更しようとししないでください。

Oracle Solaris OS ブレークの結果

このセクションで説明する動作は、次の処理を行なった場合に発生します。

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。

これらのタイプのブレーク後に次のプロンプトが表示されます。

```
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。

制御ドメインの停止またはリブートの結果

次の表に、制御 (primary) ドメインの停止またはリブートによって予想される動作を示します。

表 12-1 制御ドメインの停止またはリブートによって予想される動作

コマンド	ほかのドメインが構成されているか	動作
<code>halt</code>	未構成	ホストの電源が切断され、SP で電源が投入されるまで切断されたままです。
	構成	変数 <code>auto-boot?</code> が <code>true</code> に設定されている場合は、ソフトリセットが行われてブートします。変数 <code>auto-boot?</code> が <code>false</code> に設定されている場合は、ソフトリセットが行われて <code>ok</code> プロンプトで停止します。
<code>reboot</code>	未構成	ホストをリブートしますが、電源は切断されません。
	構成	ホストをリブートしますが、電源は切断されません。
<code>shutdown -i 5</code>	未構成	ホストの電源が切断され、SP で電源が投入されるまで切断されたままです。

表 12-1 制御ドメインの停止またはリブートによって予想される動作 (続き)

コマンド	ほかのドメインが構成されているか	動作
	構成	ソフトリセットが行われてリブートします。

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、93 ページの「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 Logical Domains 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 (SP) を使用するときの注意点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](http://www.oracle.com/technetwork/documentation/sparc-tseries-servers-252697.html)) を参照してください。

い。

既存の ILOM コマンドに、オプションを 1 つ追加できます。

```
-> set /HOST/bootmode config=config-name
```

`config=config-name` オプションを使用すると、次回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。次回のホストリセットまたは電源投入時に有効になります。

▼ ドメインの構成をデフォルトまたは別の構成にリセットする方法

- 次のコマンドを実行して、次回の電源投入時に論理ドメインの構成をデフォルトの出荷時構成にリセットします。

```
-> 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、新しい構成を読み込む必要があります。

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、マスタート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つのマスタート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つのマスタートドメインを指定しています。

```
# ldm add-domain master=apple,lemon,orange,peach pine
```

各マスタートドメインには、マスタート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、マスタート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。1つのスレーブドメインに複数のマスタートドメインが指定されている場合、最初のマスタートドメインに障害が発生すると、そのすべてのスレーブドメインに対して定義済みの障害ポリシーがトリガーされます。

注- 複数のマスタートドメインに同時に障害が発生した場合、指定された障害ポリシーのうち1つのみが、影響を受けるすべてのスレーブドメインに対して実施されます。たとえば、障害が発生したマスタートドメインに stop および panic という障害ポリシーが定義されている場合、すべてのスレーブドメインが停止するか、パニックが発生します。

マスタート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、マスタートド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無視します。
- panic は、マスタートド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、マスタートド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、マスタートド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。

この例では、マスタート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。

```
# ldm set-domain failure-policy=ignore apple
# ldm set-domain failure-policy=panic lemon
```

```
# ldm set-domain failure-policy=reset orange
# ldm set-domain failure-policy=stop peach
```

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 - 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、[262 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 18-6](#)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。

- 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=panic 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にパニックを発生させ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るか、パニックが発生します。最初に障害が発生したマスタートドメインによって、スレーブドメインの動作が決定されます。

```
# ldm add-domain failure-policy=reset twizzle
# ldm set-domain failure-policy=panic primary
# ldm add-domain master=twizzle,primary chocktaw
```

- 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ト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。

```
# ldm set-domain master=primary orange
# ldm set-domain master=orange,primary tangerine
# 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=stop
```

```
DEPENDENCY
master=primary
```

```
-----
NAME                STATE      FLAGS    UTIL
tangerine          bound     - - - - -
```

```
HOSTID
0x84f948e9
```

```
CONTROL
failure-policy=ignore
```

```
DEPENDENCY
master=orange,primary
```

- 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。

```
# 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。

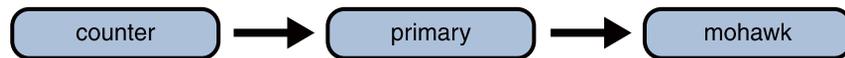
次の例は、依存サイクルがどのように作成されるかを示します。最初のコマンドは、mohawk というスレーブドメインを作成します。このドメインは、マスタードメインに primary を指定します。その結果、mohawk は、次のような依存関係の連鎖で primary に依存します。

図 12-1 単一のドメインの依存関係



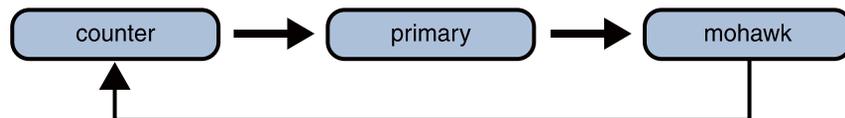
2つめのコマンドは、primary というスレーブドメインを作成します。このドメインは、マスタードメインに counter を指定します。その結果、次のような依存関係の連鎖で、mohawk が primary に依存し、primary が counter に依存します。

図 12-2 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、次のような依存サイクルが生成されます。

図 12-3 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

ドメインとそのドメイン内の仮想 CPU 番号は、特定の物理 CPU 番号に対応しており、次の手順を使用して確認できます。

▼ CPU 番号を確認する方法

- 1 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
```

- 2 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。

メモリアドレスのマッピング

ドメインとそのドメイン内の実メモリアドレスは、特定の物理メモリアドレス (PA) に対応しており、次のように確認できます。

▼ 実メモリアドレスを確認する方法

- 1 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
```

- 2 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。
ここで pa と $size$ は、その行の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 12-1 に示すような論理ドメインの構成があり、物理 CPU 番号 5 に対応するドメインと仮想 CPU、および物理アドレス $0x7e816000$ に対応するドメインと実アドレスを確認すると仮定します。

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=0x8000000|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 $0x8000000 + (0x7e816000 - 0x78000000) = 0xe816000$ です。

例 12-1 Logical Domains 構成の解析可能な長形式のリストの例

```
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=0x8000000|pa=0x80000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
```

例 12-1 Logical Domains 構成の解析可能な長形式のリストの例 (続き)

```
|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=0x8000000|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=0x8000000|pa=0x78000000|size=1073741824
...
```

ユニバーサル固有識別子の使用

Oracle VM Server for SPARC 3.0 2.0 以降のリリースでは、各ドメインに汎用一意識別子 (UUID) が割り当てられています。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 - 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 は移行されません。

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。

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 3.0 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。

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。

- 271 ページの「Oracle VM Server for SPARC P2V ツールの概要」
- 274 ページの「バックエンドデバイス」
- 275 ページの「Oracle VM Server for SPARC P2V ツールのインストール」
- 278 ページの「ldmp2v コマンドの使用」

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 10 OS または 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することができます。

- 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 が動作するが、論理ドメインでは動作していない sun4v システム

注 - `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で実行します。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せん。
- 変換フェーズ。ターゲットシステムの制御ドメインで実行します。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各フェーズで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。

h、/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-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

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 - `ldmp2v` は `SUNWldmp2v` パッケージから Oracle Solaris 10 システムにインストールされ、`ldmp2v` はデフォルトで `ldomsmanager` パッケージから Oracle Solaris 11 システムにインストールされます。

準備すべき事柄

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が、次のいずれかの OS で Logical Domains 1.1 以降を実行している。
 - Solaris 10 10/08 OS
 - Solaris 10 5/08 OS と適切な Logical Domains 1.1 パッチ
- ゲストドメインが、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。

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交換できます。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。ゲストドメインでこれらの手順を実行する詳細については、『[Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management](#)』の「[Migrating a Non-Global Zone to a Different Machine](#)」を参照してください。

注-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ません。さらに、このツールはゾーンパスのストレージを移動したり、構成したりすることはありません。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。『[Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management](#)』の「[Migrating a Non-Global Zone to a Different Machine](#)」を参照してください。

▼ Oracle VM Server for SPARC P2V ツールのインストール方法

次の手順では、SUNWldmp2v パッケージを使用して、ldmp2v コマンドを Oracle Solaris 10 システムにインストールする方法を説明します。

ldmp2v コマンドを Oracle Solaris 11 システムにインストールする場合、ldomsmanager パッケージをインストールするとデフォルトでこのコマンドがインストールされます。

- 1 Oracle VM Server for SPARC のダウンロードページ <http://www.oracle.com/virtualization/index.html> に移動します。
- 2 P2V ソフトウェアパッケージ `SUNWldmp2v` をダウンロードします。
`SUNWldmp2v` パッケージは、Oracle VM Server for SPARC zip ファイルに入っています。
- 3 管理者になります。
Oracle Solaris 10 の場合は、『[System Administration Guide: Security Services](#)』の「[Configuring RBAC \(Task Map\)](#)」を参照してください。Oracle Solaris 11.1 の場合は、『[Oracle Solaris 11.1 Administration: Security Services](#)』のパート III 「[Roles, Rights Profiles, and Privileges](#)」を参照してください。
- 4 `pkgadd` コマンドを使用して、`SUNWldmp2v` パッケージをインストールします。

```
# pkgadd -d . SUNWldmp2v
```
- 5 `/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場合は BACKEND_SPARSE="yes"、空白でないボリュームまたはファイルの場合は BACKEND_SPARSE="no"
 - 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の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、3つのフェーズの例が含まれます。

例13-1 収集フェーズの例

ldmp2v collect コマンドの使用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/ctl1d0s0    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/ctl1d0s3    8258597   4304   8171708     1%      /var
swap                   4487448   16   4487432     1%      /var/run
swap                   4487448   16   4487432     1%      /tmp
/dev/dsk/ctl0d0s0    1016122    9   955146      1%      /u01
vandikhout:/u1/home/dana
6230996752 1051158977 5179837775    17%    /home/dana
```

次の例は、ソースシステムとターゲットシステムが1つのNFS マウント済み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ています。

```
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。Flash ユーティリティは、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。

例 13-1 収集フェーズの例 (続き)

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。NetBackup などのサードパーティー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がすでに使用可能である場合は、none アーカイブメソッドを使用して、ファイルシステム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 13-2 準備フェーズの例

ldmp2v prepare コマンドの使用方法的例を次に示します。

- 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
```

例 13-2 準備フェーズの例 (続き)

```

DISK
  NAME    DEVICE  TOUT  MPGROUP          VOLUME                                SERVER
  disk0   disk0   10000  mp0               volumia-vol0@primary-vds0
  disk1   disk1   10000  mp1               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できることを示しています。

```

# 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できることを示しています。

```

# 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 13-3 変換フェーズの例

`ldmp2v convert` コマンドの使用方法の例を次に示します。

- 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が、

例 13-3 変換フェーズの例 (続き)

```

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 の使用方法の詳細については、『[Oracle Solaris 10 8/11 Installation Guide: Custom JumpStart and Advanced Installations](#)』を参照してください。

注 - 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 '^]'.

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

```

例 13-3 変換フェーズの例 (続き)

```
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。

例 13-3 変換フェーズの例 (続き)

注 -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
```

例 13-3 変換フェーズの例 (続き)

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 Configuration Assistant (Oracle Solaris 10)

Oracle VM Server for SPARC Configuration Assistant (`ldmconfig` コマンド) を使用すると、基本的なプロパティを設定することによって論理ドメインの構成手順を実行できます。Oracle VM Server for SPARC Configuration Assistant は、チップマルチスレッディング (CMT) をベースとするシステムで実行されます。

Configuration Assistant は、構成データを収集したあと、論理ドメインとしてブートするのに適した構成を作成します。Configuration Assistant によって選択されるデフォルト値を使用して、有効なシステム構成を作成することもできます。

注 - `ldmconfig` コマンドは Oracle Solaris 10 システムでのみサポートされます。

この章のほかに、[ldmconfig\(1M\)](#) マニュアルページを参照してください。

Configuration Assistant (`ldmconfig`) の使用

`ldmconfig` コマンドでは、ユーザーインタフェース画面に一致する一連の操作が実行されます。最終的には、論理ドメインに配備可能な構成が作成されます。

次のセクションでは、`ldmconfig` コマンドをインストールする方法および Configuration Assistant ツールの機能の一部について説明します。

Configuration Assistant のインストール

Configuration Assistant は、`SUNWldm` パッケージの一部として提供されます。

`SUNWldm` パッケージをインストールすると、`/usr/sbin` ディレクトリに `ldmconfig` コマンドが格納されます。このコマンドは、旧バージョンでの使用のために、`/opt/SUNWldm/bin` ディレクトリにもインストールされます。

準備すべき事柄

Configuration Assistant をインストールして実行する前に、次の条件を満たしていることを確認してください。

- ターゲットシステムで Logical Domains 1.2 ソフトウェア以降が実行されている。
- 端末ウィンドウに、1 行あたり 80 文字以上で 24 行表示できる。

制限事項および既知の問題

Configuration Assistant には、次の制限事項があります。

- ldmconfig を使用しながら端末のサイズを変更すると、文字化けが発生することがある
- UFS ディスクファイルは仮想ディスクとしてのみサポートされる
- 既存の論理ドメイン構成が存在しないシステムのみで機能する
- 仮想コンソール端末集配装置のポートは 5000 - 5100
- ゲストドメイン、サービス、およびデバイスに使用されるデフォルトの名前は変更不可

ldmconfig の機能

ldmconfig コマンドでは、ユーザーインタフェース画面に一致する一連の操作が実行されます。最後の手順に到達するまで、後方(前の手順)および前方(次の手順)に移動できます。最後の手順では、構成が生成されます。いつでも Configuration Assistant を終了したり、構成をリセットしてデフォルトを使用できます。最後の画面では、論理ドメインに構成を配備できます。

まず、Configuration Assistant は、システムを自動的に検査し、ベストプラクティスに基づいて最適なデフォルトのプロパティを判断してから、これらのプロパティのうち配備の制御に必要なプロパティを表示します。これは完全なリストではないことに注意してください。他のプロパティを設定して構成をさらにカスタマイズできます。

ldmconfig ツールの用法については、[ldmconfig\(1M\)](#) マニュアルページを参照してください。

次のプロパティを調整できます。

- ゲストドメインの数。作成するアプリケーションのゲストドメインの数を指定します。ゲストドメイン数の最小値は1です。最大値は、使用できる VCPU リソースによって決まります。たとえば、64 スレッドの CMT システムで、制御ドメイン用に4つのスレッドを予約し、各ゲストドメインに1つのスレッドを使用して最大 60 個のゲストドメインを作成できます。ベストプラクティスが選択されている場合、ゲストドメインあたりの VCPU リソースの最小数は、1 コアになります。そのため、1 コアあたり 8 スレッドの 8 コアシステムでベストプラクティスが選択されている場合、それぞれ1つのコアが割り当てられた最大7個のゲストドメインを作成できます。また、制御ドメインにも1つのコアが割り当てられます。

Configuration Assistant は、そのシステムに構成可能なドメインの最大数を表示します。

Configuration Assistant は次のタスクを実行し、ドメインを作成します。

- すべてのドメインに対して実行するタスク
 - 5000 - 5100 のポートに仮想端末サービスを作成
 - 仮想ディスクサービスを作成
 - 指定されたネットワークアダプタに仮想ネットワークスイッチを作成
 - 仮想端末サーバーデーモンを有効化
- 各ドメインに対して実行するタスク
 - 論理ドメインを作成
 - ドメインに割り当てられる VCPU を構成
 - ドメインに割り当てられるメモリーを構成
 - 仮想ディスクとして使用する UFS ディスクファイルを作成
 - ディスクファイルの仮想ディスクサーバーデバイス (vdsdev) を作成
 - ディスクファイルをドメインの仮想ディスク `vdisk0` として割り当て
 - 指定されたネットワークアダプタの仮想スイッチに接続された仮想ネットワークアダプタを追加
 - OBP プロパティ `auto-boot?=true` を設定
 - OBP プロパティ `boot-device=vdisk0` を設定
 - ドメインをバインド
 - ドメインを起動
- デフォルトのネットワーク。新しいドメインで仮想ネットワークに使用するネットワークアダプタを指定します。このアダプタは、システムに存在する必要があります。Configuration Assistant は、現在システムによってデフォルトアダプタとして使用されているアダプタ、およびリンクステータスがアクティブになっているアダプタ (ケーブル接続されているアダプタ) を強調表示します。

- 仮想ディスクサイズ。新しい各ドメインの仮想ディスクを作成します。これらの仮想ディスクは、ローカルファイルシステムに存在するディスクファイルに基づいて作成されます。このプロパティーは、各仮想ディスクのサイズをGバイト単位で制御します。最小サイズの8Gバイトは、Oracle Solaris 10 OSを格納するために必要なおおよそのサイズに基づいています。最大サイズは100Gバイトです。

Configuration Assistant がすべてのドメインのディスクファイルを格納するのに十分な領域のあるファイルシステムを検出できない場合、エラー画面が表示されます。この場合、アプリケーションを再実行する前に次の操作が必要になることがあります。

- 仮想ディスクのサイズを減らす
- ドメインの数を減らす
- より容量の大きいファイルシステムを追加する
- 仮想ディスクディレクトリ。新しいドメインの仮想ディスクとして作成するファイルを格納するために十分な容量のあるファイルシステムを指定します。このディレクトリは、選択するドメインの数、および仮想ディスクのサイズに基づいて指定します。これらのプロパティーの値が変更された場合は、値を再計算して格納先ディレクトリを選択する必要があります。Configuration Assistant は、十分な領域のあるファイルシステムのリストを表示します。ファイルシステム名を指定すると、このファイルシステムに `/ldoms/disks` というディレクトリが作成され、このディレクトリにディスクイメージが作成されます。
- ベストプラクティス。プロパティー値にベストプラクティスを使用するかどうかを指定します。
 - `yes` という値を選択すると、Configuration Assistant によっていくつかの構成プロパティー値にベストプラクティスが使用されます。ベストプラクティスでは、最小値として、ドメインあたり1コアという値が適用されます。これにはシステムドメインも含まれます。その結果、ゲストドメインの最大数は、システムに存在するコアの合計数から、システムドメイン用の1コアを引いた数に制限されます。たとえば、それぞれ8つのコアが割り当てられた2ソケット SPARC Enterprise T5140 の場合、ゲストドメインの最大数はシステムドメインを除いた15個となります。
 - `no` という値を選択すると、Configuration Assistant によって、最少で1スレッドが割り当てられたドメインの作成が許可されます。ただし、システムドメインのスレッド数は4以上に保持されます。

次に、Configuration Assistant は、作成される配備構成の概略を表示します。これには次の情報が含まれます。

- ドメイン数
- 各ゲストドメインに割り当てられるCPU
- 各ゲストドメインに割り当てられるメモリー
- 仮想ディスクのサイズおよび場所
- ゲストドメインの仮想ネットワークサービスに使用されるネットワークアダプタ

- システムによってサービスに使用される CPU およびメモリーの量
- 有効な Oracle Solaris OS DVD が識別されると、これを使用して共有仮想 CD-ROM デバイスが作成され、ゲストドメインに Oracle Solaris OS をインストールできるようになります。

最後に、Configuration Assistant はシステムを構成して、指定された Logical Domains 配備を作成します。また、実行される処理についての説明と、システムを構成するために実行するコマンドを表示します。この情報は、システムを構成するために必要な ldm コマンドの使用法を理解するのに役立ちます。



注意 - この構成手順に影響を与えたり、このプロセスを中断したりしないでください。システムの構成が不完全になることがあります。

コマンドが正常に終了したら、変更を有効にするためにシステムをリブートしてください。

電源管理の使用

この付録では、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。

- パフォーマンス。システムは、使用可能なすべての電力を使用できます。
- エラスティック。システムの電力使用量は、現在の使用率のレベルに合わせて変化します。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。

PMの機能は以下のとおりです。

- **CPUコアの自動的な無効化。** エラスティックポリシーが有効になっている場合、CPUコアのすべてのスレッド(ストランド)が無効になっているときに、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。** Oracle VM Server for SPARC 2.0以降のリリースでは、Logical Domains ManagerはCPUクロックサイクルのスキップを自動的に調整できます。この機能はSPARC T3およびSPARC T4プラットフォームでのみ使用できます。調整により、スキップされるクロックサイクル数が増減し、すべてのドメインを電力使用率のしきい値内に保つことができます。PMは、CPU使用率に基づいて、このような調整を行うかどうかを決定します。システムがパフォーマンスポリシーを使用する場合、スキップされるクロックサイクルの数は自動的に0に調整されます。
- **ディープアイドルモードでのメモリ操作。** Oracle VM Server for SPARC 2.0以降のリリースでは、SPARC T3およびSPARC T4プラットフォームでエラスティックポリシーが有効にされている場合、使用率の低いメモリーが自動的に、電力節減のためより深いアイドルモードで動作するように構成されます。
- **電力の制限。** SPARC T3およびSPARC T4プラットフォームで、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手法を使用して電力が削減されます。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、電源管理が可能なリソース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。

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、ドメイン構成に対する一部の変更においては、電力の制限を超えていないことの検証が最初に行われます。電力の制限を超えている場合、要求されたリソースの一部についてのみ変更または追加が行われる可能性があります。あとで電力の制限が増やされた場合、変更失敗したリソースをそのあとで追加できます。

ドメインの負荷によりリソースの消費電力が増えた場合、消費電力が電力の制限を下回っているリソースのみが正常に電力投入されます。

- **Solaris Power Aware Dispatcher (PAD)**。Oracle VM Server for SPARC 3.0 以降のリリースでは、Oracle Solaris 11.1 OS が実行されているゲストドメインは、SPARC T3 および SPARC T4 システム上で Power Aware Dispatcher (PAD) を使用して、アイドル状態や使用率が低いリソースの消費電力を最小限にします。Logical Domains Manager の代わりに、PAD が CPU クロックサイクルのスキップ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。

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 (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の実行と、簡易ネットワーク管理プロトコル (Simple Network Management Protocol、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。

- 296 ページの「Oracle VM Server for SPARC 管理情報ベースの概要」
- 299 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」
- 303 ページの「セキュリティの管理」
- 305 ページの「ドメインのモニタリング」
- 328 ページの「SNMP トラップの使用」
- 336 ページの「ドメインの起動と停止」

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)

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。

- 296 ページの「ソフトウェアコンポーネント」
- 297 ページの「システム管理エージェント」
- 298 ページの「Logical Domains Manager と Oracle VM Server for SPARC MIB」
- 298 ページの「Oracle VM Server for SPARC MIB オブジェクトツリー」

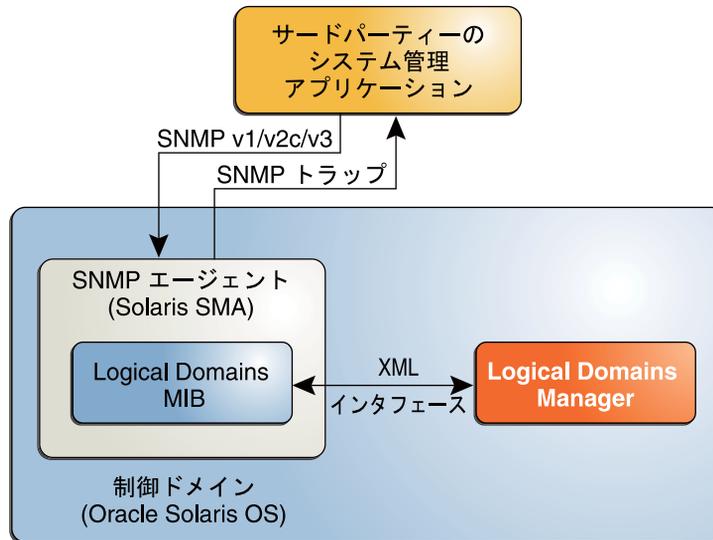
ソフトウェアコンポーネント

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、297 ページの「システム管理エージェント」と298 ページの「Logical Domains Manager と Oracle VM Server for SPARC MIB」で説明しています。

図 16-1 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

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 16-1 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。

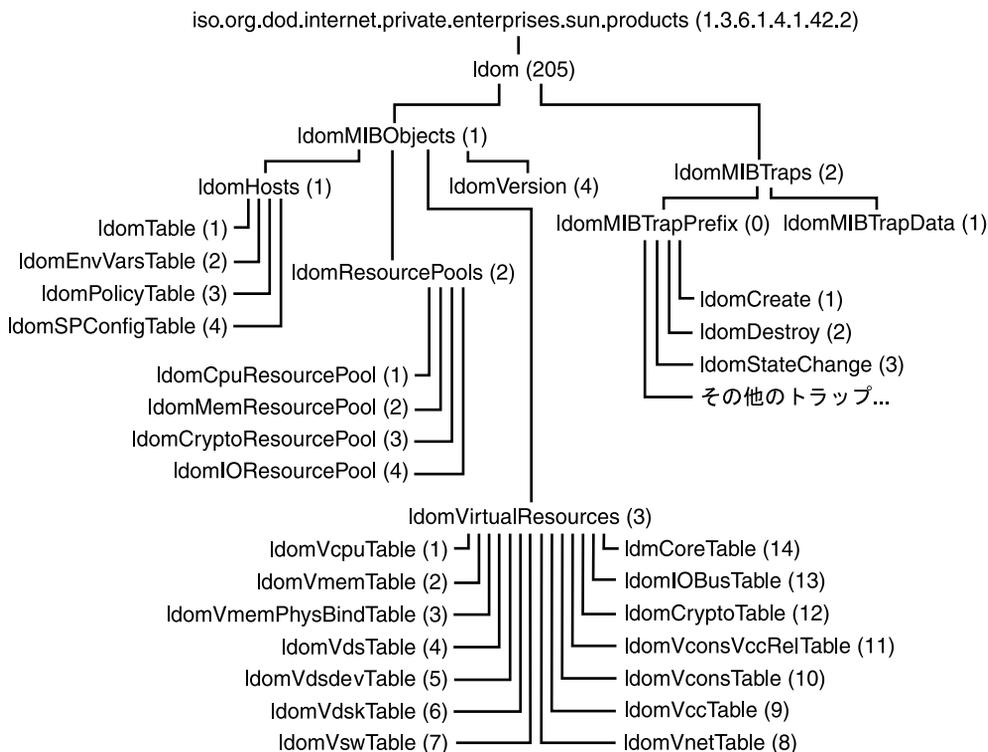
Oracle VM Server for SPARC MIB オブジェクトツリー

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 16-2 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 Solaris 10 および Oracle Solaris 11 システムへの 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 (タスクマップ)

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。	pkgadd コマンドを使用して、Oracle Solaris 10 システムに SUNWldmib.v パッケージをインストールします。または、pkg install コマンドを使用して、Oracle Solaris 11 システムに system/ldoms/mib パッケージをインストールします。	300 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。	301 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。	pkgrm コマンドを使用して、Oracle Solaris 10 システムから SUNWldmib パッケージを削除します。または、pkg remove コマンドを使用して、Oracle Solaris 11 システムから system/ldoms/mib パッケージを削除します。	302 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 Solaris 10 および Oracle Solaris 11 システムに 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 3.0 ソフトウェアの一部として提供されています。

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。301 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 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 3.0 ソフトウェアをダウンロードしてインストールします。第2章「ソフトウェアのインストールおよび有効化」を参照してください。

- 1 システムで **Oracle Solaris 10 OS** または **Oracle Solaris 11 OS** が実行されているかどうかを確認します。

```
# uname -r
```

- 2 **primary** ドメインに Oracle VM Server for SPARC MIB ソフトウェアをインストールします。

- Oracle Solaris 10 MIB ソフトウェアパッケージ **SUNWldmib** をインストールします。

```
# pkgadd -d . SUNWldmib.v
```

- Oracle Solaris 11 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。

```
# pkg install -v -g IPS-package-directory/ldoms.repo mib
```

▼ 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 システムで **Oracle Solaris 10 OS** または **Oracle Solaris 11 OS** が実行されているかどうかを確認します。

```
# uname -r
```

2 SNMP 構成ファイルを更新します。

- Oracle Solaris 10 SNMP エージェント SNMP 構成ファイルを更新します。

次の行を `/etc/sma/snmp/snmpd.conf` 構成ファイルに追加します。

```
dload ldomMIB /opt/SUNWldmib/lib/ldomMIB.so
```

- Oracle Solaris 11 SNMP エージェント 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
```

3 SMF サービスを再起動します。

- Oracle Solaris 10 SNMP エージェント サービスを再起動します。

```
# svcadm restart svc:/application/management/sma:default
```

- Oracle Solaris 11 SNMP エージェント サービスを再起動します。

```
# 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 Solaris 10 または Oracle Solaris 11 システムから 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 モジュールを読み込み解除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。

- Oracle Solaris 10 SNMP エージェント サービスを停止します。

```
# svcadm disable svc:/application/management/sma:default
```

- Oracle Solaris 11 SNMP エージェント サービスを停止します。

```
# 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。

- Oracle Solaris 10 MIB ソフトウェアパッケージを削除します。

```
# pkgrm SUNWldmib
```

- Oracle Solaris 11 MIB ソフトウェアパッケージを削除します。

```
# pkg uninstall system/ldoms/mib
```

- 3 SNMP 構成ファイルを更新します。
 - Oracle Solaris 10 SNMP エージェント 構成ファイルを更新します。
インストール中に /etc/sma/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
```
 - Oracle Solaris 11 SNMP エージェント 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。
 - Oracle Solaris 10 SNMP エージェント サービスを再起動します。

```
# svcadm restart svc:/application/management/sma:default
```
 - Oracle Solaris 11 SNMP エージェント サービスを再起動します。

```
# 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 - 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。この種類のユーザーは、Logical Domains Manager 用 Oracle Solaris の役割に基づくアクセス制御 (RBAC) 機能を使用して構成した可能性のあるユーザーとやり取りしたり競合したりすることはありません。

▼ 最初の snmpv3 ユーザーを作成する方法

この手順では、Oracle Solaris 10 または Oracle Solaris 11 システムで最初の 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、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。

- Oracle Solaris 10 SNMP エージェントを停止します。

```
# svcadm disable -t svc:/application/management/sma:default
```

- Oracle Solaris 11 SNMP エージェントを停止します。

```
# 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-パスワードは、8文字以上である必要があります。

- Oracle Solaris 10 システムに初期ユーザーを作成します。

```
# /usr/sfw/bin/net-snmp-config --create-snmpv3-user -a my-password initial-user
```

- Oracle Solaris 11 システムに初期ユーザーを作成します。

```
# /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。

- Oracle Solaris 10 SNMP エージェントを起動します。

```
# svcadm enable svc:/application/management/sma:default
```

- Oracle Solaris 11 SNMP エージェントを起動します。

```
# 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。

```
# 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。

- 305 ページの「環境変数の設定」
- 306 ページの「Oracle VM Server for SPARC MIB の問い合わせ」
- 308 ページの「Oracle VM Server for SPARC MIB 情報の取得」

環境変数の設定

▼ 環境変数を設定する方法

Oracle VM Server for SPARC MIB に問い合わせる前に、使用するシェルの環境変数を設定する必要があります。Oracle Solaris 10 と Oracle Solaris 11 では値が異なります。この手順では、C シェル、Bourne シェル、および Korn シェルの環境変数を設定する方法を説明します。

1 Oracle Solaris 10 の場合、PATH、MIBDIRS、および MIBS 環境変数を設定します。

■ C シェルユーザーの場合

```
% setenv PATH /usr/sfw/bin:$PATH
% setenv MIBDIRS /opt/SUNWldmib/lib/mibs:/etc/sma/snmp/mibs
% setenv MIBS +SUN-LDOM-MIB
```

■ Bourne および Korn シェルユーザーの場合

```
$ PATH=/usr/sfw/bin:$PATH; export PATH
$ MIBDIRS=/opt/SUNWldmib/lib/mibs:/etc/sma/snmp/mibs; export MIBDIRS
$ MIBS+=SUN-LDOM-MIB; export MIBS
```

2 Oracle Solaris 11 の場合、PATH、MIBDIRS、および MIBS 環境変数を設定します。

■ C シェルユーザーの場合

```
% setenv PATH /usr/bin:$PATH
% setenv MIBDIRS /opt/SUNWldmib/lib/mibs:/etc/net-snmp/snmp/mibs
% setenv MIBS +SUN-LDOM-MIB
```

■ Bourne および Korn シェルユーザーの場合

```
$ 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

▼ Oracle VM Server for SPARC MIB オブジェクトを取得する方法

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。

● 1つ以上の MIB オブジェクトを取得します。

■ 単一の MIB オブジェクトを取得します。

```
# snmpget -v version -c community-string host MIB-object
```

■ 多数の MIB オブジェクトを取得します。

`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
```

例 16-1 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 16-2 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::ldomThreading.1 = STRING: max-throughput
SUN-LDOM-MIB::ldomThreading.2 = STRING: max-throughput
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

```

- 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 16-3 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 16-1 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。 <ul style="list-style-type: none"> ■ 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。 <ul style="list-style-type: none"> ■ 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。

表 16-1 ドメインテーブル (ldomTable) (続き)

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。 <ul style="list-style-type: none"> ■ 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。

表 16-1 ドメインテーブル (ldomTable) (続き)

名前	データ型	アクセス	説明
ldomThreading	表示文字列	読み取り専用	<p>サイクルあたりの命令数 (IPC) が高いスレッド制御をドメインに指定します。動的スレッドによって、1 コアあたりにアクティブにするハードウェアスレッドの数が制御されません。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> ■ max-throughput、1 コアあたりすべてのストランドがアクティブであることを意味します (デフォルト値) ■ max-ipc、1 コアあたり1つのストランドがアクティブであることを意味します
ldomWholeCore	整数	読み取り専用	<p>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。</p>
ldomCpuArch	表示文字列	読み取り専用	<p>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。</p> <ul style="list-style-type: none"> ■ 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	<p>ゲストドメインの停止グループ番号。SPARC64-X システムでは、順序付けられた停止が SP で開始されると、停止グループ番号の降順 (15 から 0 の順番) でドメインが停止されません。デフォルト値は 15 です。</p>

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 16-2 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 16-3 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

表 16-3 ドメインポリシーテーブル (ldomPolicyTable) (続き)

名前	データ型	アクセス	説明
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 16-4 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

表 16-4 サービスプロセッサ構成テーブル (ldomSPConfigTable) (続き)

名前	データ型	アクセス	説明
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)

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 16-5 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
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。 <ul style="list-style-type: none"> ■ 1 は MHz ■ 2 は GHz デフォルト値は MHz です。

表 16-6 メモリー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

表 16-6 メモリーリソースプールのスカラー変数 (続き)

名前	データ型	アクセス	説明
<code>ldomMemRpReserved</code>	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、 <code>MemRpReservedUnit</code> のメモリー容量
<code>ldomMemRpCapacityUnit</code> および <code>ldomMemRpReservedUnit</code>	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。 <ul style="list-style-type: none"> ■ 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 16-7 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
<code>ldomCryptoRpCapacity</code>	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
<code>ldomCryptoRpReserved</code>	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 16-8 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
<code>ldomIOBusRpCapacity</code>	整数	読み取り専用	プールが許可する最大予約数
<code>ldomIOBusRpReserved</code>	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

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 16-9 仮想 CPU テーブル (`ldomVcpuTable`)

名前	データ型	アクセス	説明
<code>ldomVcpuLdomIndex</code>	整数	読み取り専用	仮想 CPU を含むドメインを表す <code>ldomTable</code> へのインデックスとして使用される整数

表 16-9 仮想 CPU テーブル (ldomVcpuTable) (続き)

名前	データ型	アクセス	説明
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 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。

表 16-9 仮想 CPU テーブル (ldomVcpuTable) (続き)

名前	データ型	アクセス	説明
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 16-10 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 16-11 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 16-12 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 <i>service-name</i> 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 16-13 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

表 16-13 仮想ディスクサービスデバイステーブル (ldomVdsdevTable) (続き)

名前	データ型	アクセス	説明
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的な名前を指定します。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、 <code>ldm add-vdsdev</code> 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、 <code>ldm add-vdsdev</code> 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 16-14 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 <code>ldomTable</code> 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 <code>ldomVdsdevTable</code> 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
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、 <code>ldm add-vdisk</code> コマンドによって指定される <i>disk-name</i> です。

表 16-14 仮想ディスクテーブル (ldomVdiskTable) (続き)

名前	データ型	アクセス	説明
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。

- 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 16-3 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 16-15 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

表 16-15 仮想スイッチサービステーブル (ldomVswTable) (続き)

名前	データ型	アクセス	説明
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 16-16 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

表 16-16 仮想ネットワークデバイステーブル (ldomVnetTable) (続き)

名前	データ型	アクセス	説明
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用

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 16-17 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。

表 16-18 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。
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。

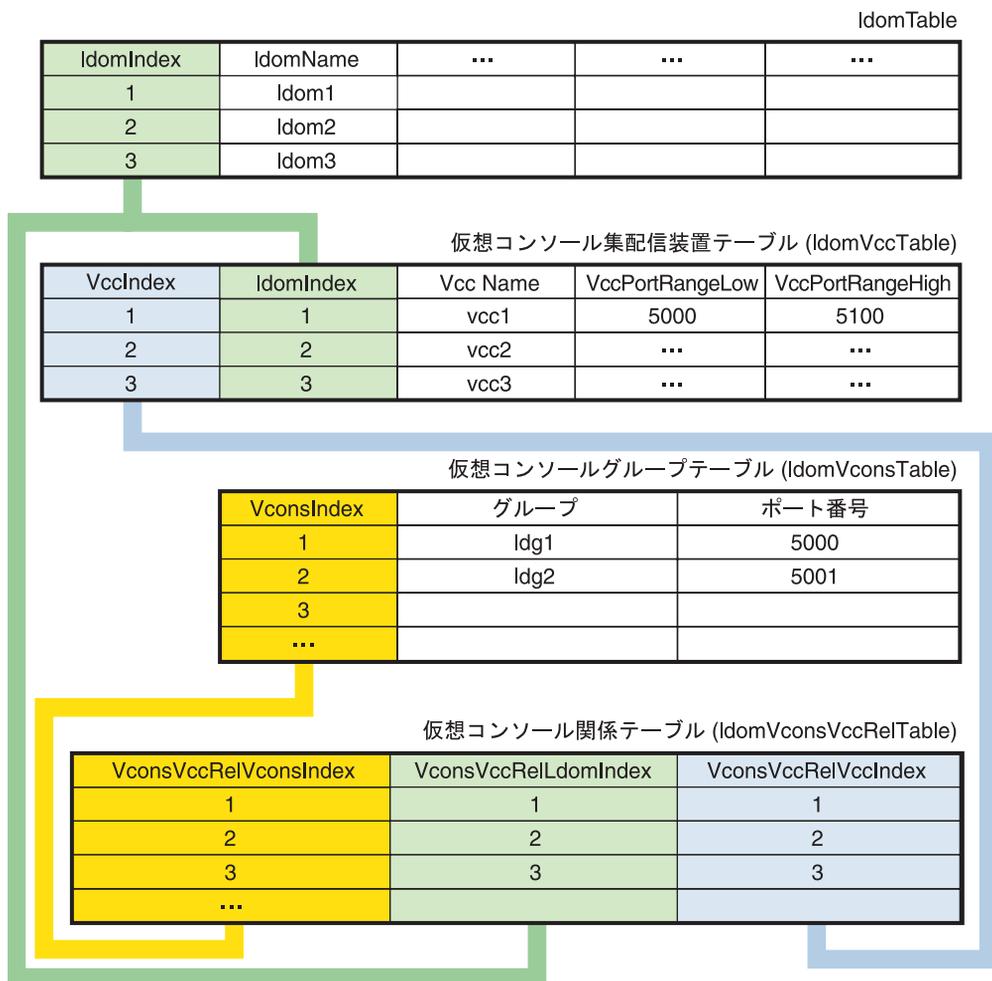
表 16-19 仮想コンソール関係テーブル (ldomVconsVccRelTable)

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 16-4 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 (IdomCryptoTable)

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、モジュラー演算ユニット (MAU) と呼ばれる場合があります。

表 16-20 暗号化装置テーブル (IdomCryptoTable)

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

表 16-20 暗号化装置テーブル (ldomCryptoTable) (続き)

名前	データ型	アクセス	説明
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 16-21 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

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 16-22 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

表 16-22 コアテーブル (ldomCoreTable) (続き)

名前	データ型	アクセス	説明
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

Logical Domains 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 Logical Domains バージョンをサポートしています。Oracle VM Server for SPARC MIB には、Logical Domains 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 16-23 Logical Domains 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。

Oracle VM Server for SPARC MIB では、Oracle Solaris 10 と Oracle Solaris 11 の両方に同じ SNMP トラップが提供されます。

ただし、snmpttrapd デーモンでは Oracle Solaris 11 用の受信トラップのすべてを必ずしも自動的に受け入れなくなりました。その代わりに、承認された SNMP v1 と v2c コ

コミュニティー文字列、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

トラップの送受信

Oracle Solaris 11 MIB では、Oracle Solaris 10 MIB で提供されるものと同じ SNMP トラップが提供されます。ただし、net-snmp のバージョンが異なるため、異なる方法で構成する必要があります。Oracle Solaris 10 MIB では、snmptrapd がすべての受信通知を受け入れ、自動的に記録します。Oracle Solaris 11 MIB では、受信通知に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。

- Oracle Solaris 10 システムでトラップを構成します。

/etc/sma/snmp/snmpd.conf 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。

```
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。

- Oracle Solaris 11 システムで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。

```
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ー文字列として `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。

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ー `snmptrapd` を起動します。

例 16-4 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。次の指令を含めて Oracle Solaris 10 `/etc/sma/snmp/snmpd.conf` ファイルまたは Oracle Solaris 11 `/etc/net-snmp/snmp/snmpd.conf` ファイルを更新します。

```
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ーを起動します。
 - Oracle Solaris 10 SNMP トラップデーモンユーティリティーを起動します。

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` ユーティリティーは、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/sfw/sbin/snmptrapd -P -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
TRAP from localhost on 5/18/2007 at 16:30:10 Enterprise=. Type=0 SubType=0
with Varbinds: DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (47105)
0:07:51.05 SNMPv2-MIB::snmpTrapOID.0 = OID: SUN-LDOM-MIB::ldomCreate
SUN-LDOM-MIB::ldomIndexNotif = INTEGER: 3 SUN-LDOM-MIB::ldomName = STRING: ldg2
```

```
SUN-LDOM-MIB::ldomTrapDesc = STRING: Ldom Created
Security info:TRAP2, SNMP v2c, 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 Solaris 11 SNMP トラップデーモンユーティリティーを起動します。**
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 ユーティリティーは、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 16-24 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 16-25 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 16-26 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 の数に変更されたときに通知を行います。

表 16-27 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に変更されたときに通知を行います。

表 16-28 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
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。 <ul style="list-style-type: none"> ■ 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。 <ul style="list-style-type: none"> ■ 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 16-29 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

表 16-29 ドメインの仮想ディスクサービスの変更トラップ (ldomVdsChange) (続き)

名前	データ型	説明
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。 <ul style="list-style-type: none"> ■ 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 16-30 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。 <ul style="list-style-type: none"> ■ 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 16-31 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

表 16-31 仮想スイッチの変更トラップ (ldomVswChange) (続き)

名前	データ型	説明
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。 <ul style="list-style-type: none"> ■ 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 16-32 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。 <ul style="list-style-type: none"> ■ 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 16-33 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

表 16-33 仮想コンソール端末集配信装置の変更トラップ (ldomVccChange) (続き)

名前	データ型	説明
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。 <ul style="list-style-type: none"> ■ 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 16-34 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。 <ul style="list-style-type: none"> ■ 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

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 16-1 を参照してください。

ドメインの起動と停止

▼ 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::ldomThreading.1 = STRING: max-throughput
SUN-LDOM-MIB::ldomThreading.2 = STRING: max-throughput
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
```

- 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 *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 コマンドを使用します。

```
# ldm list domain-name
```
 - snmpget コマンドを使用します。

```
# snmpget -v version -c community-string hostname SUN-LDOM-MIB::ldomOperState.n
```

例 16-5 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 コマンドを使用して LdomMibTest_1 ドメインの状態を取得する代わりに、snmpget コマンドを使用できます。

```
# 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::ldomThreading.1 = STRING: max-throughput
SUN-LDOM-MIB::ldomThreading.2 = STRING: max-throughput
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

```

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 *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` コマンドを使用します。

```
# ldm list domain-name
```

- `snmpget` コマンドを使用します。

```
# snmpget -v version -c community-string hostname SUN-LDOM-MIB::ldomOperState.n
```

例 16-6 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

```


Logical Domains Manager 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。

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。
[341 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、次の値が含まれる `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 ソフトウェアとやり取り可能な 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。

- 345 ページの「XML トランスポート」
- 346 ページの「XML プロトコル」
- 351 ページの「イベントメッセージ」
- 356 ページの「Logical Domains Manager の処理」
- 358 ページの「Logical Domains Manager のリソースとプロパティ」
- 373 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。リモート接続を切断するには、`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 - 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 LDom Manager 自身の間の通信チャン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 LDom 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 ハンドシェイクを行います。具体的には、TLS を介したセキュアチャンネルの設定後の 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 ハンドシェイクを 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 Logical Domains 定義の XML メッセージが送信されます。XML メッセージには、次の 2 つの一般的なタイプがあります。

- <LDM_interface> タグを使用する要求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

Logical Domains 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処理に基づいて 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。基本的な XML コマンドの構造は次のとおりです。

例 18-1 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>LDM Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
          </Section>
          <!-- Note: More Sections sections can be placed here -->
        </Content>
      </Envelope>
    </data>
    <!-- Note: More Data sections can be placed here -->
  </cmd>
  <!-- Note: More Commands sections can be placed here -->
</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 18-1 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。

- remove-domain サブコマンドには、-a オプションを使用できます。
- bind-domain サブコマンドには、-f オプションを使用できます。
- add-vdsdev サブコマンドには、-f オプションを使用できます。
- cancel-operation サブコマンドには、migration または reconf オプションを使用できます。
- add-spconfig サブコマンドには、-r autosave-name オプションを使用できます。
- remove-spconfig サブコマンドには、-r オプションを使用できます。
- list-spconfig サブコマンドには、-r [autosave-name] オプションを使用できません。
- stop-domain サブコマンドでは、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。

次の例の XML 部分は、リポートメッセージを含むリポートリクエストを stop-domain サブ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。データ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 (Logical Domains 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。

Logical Domains 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> セクションの内側の位置に指定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。

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るように、さらに 2 つの OVF タイプが定義されています。

- <gprop:GenericProperty> タグ
- <Binding> タグ

<gprop:GenericProperty> タグは、OVF 仕様には定義がないオブジェクトのプロパティを取り扱うために定義されました。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 サブ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> セクションでは、例 18-2 に示すようなステータスおよびメッセージ情報が提供されます。基本的な XML 要求に対する応答の構造は、次のとおりです。

例 18-2 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>
```

例 18-2 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 (続き)

```

<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> sections can be placed here -->
    </Content>
  </Envelope>
  <response>
    <status>success or failure</status>
    <resp_msg>Reason for failure</resp_msg>
  </response>
</data>
<!-- Note: More Data sections can be placed here -->
<response>
  <status>success or failure</status>
  <resp_msg>Reason for failure</resp_msg>
</response>
</cmd>
<!-- Note: More Command sections can be placed here -->
<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。ここでは、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がわかります。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。349 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、353 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。347 ページの「<LDM_interface> タグ」を参照してください。処理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 18-3 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>
```

例 18-3 イベントの登録要求メッセージの例 (続き)

```
</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 18-4 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>
</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。メッセージの処理タグは、イベントをトリガーするために実行された処理です。メッセージのデータ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 18-5 <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>
```

例 18-5 <LDM_event> 通知の例 (続き)

```
</Item>
</Section>
</Envelope>
</data>
</cmd>
</LDM_event>
```

イベントタイプ

次に、登録できるイベントのタイプを示します。

- 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処理を記述します。次に、<LDM_event> メッセージの <action> タグに指定することができるドメインイベントを示します。

- `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 データセクションにイベントが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。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。これらのイベントの登録を解除するには、処理タグを **unreg-domain-events** に設定した <LDM_interface> メッセージが必要です。

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せん。Oracle VM Server for SPARC ソフトウェアの場合、実行できるハードウェア変更は、ユーザーが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 データセクションにイベントが発生している 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。これらのイベントの登録を解除するには、<action> タグを **unreg-hardware-events** に設定した <LDM_interface> メッセージが必要です。

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 データ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。これらのイベントの登録を解除するには、<action> タグを **unreg-hardware-events** に設定した <LDM_interface> メッセージが必要です。

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データセクションには、OVF データ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。

次に、<LDM_event> メッセージの <action> タグに指定することができるイベントを示します。

- 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 データセクションに、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 の処理

<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 - 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-udpcc
- add-udpccs
- 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

ここでは、Logical Domains Manager のリソースと、リソースごとに定義できるプロパティを示します。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、Logical Domains Manager の処理の入力を作成する場合に使用できます。ただし、ドメイン移行の出力は例外です。372 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

例 18-6 ldom_info の XML 出力の例

```
<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:GenericPropertykey="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">dt90-319</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:GenericPropertykey="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:GenericPropertykey="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:GenericPropertykey="hostid">` タグ。ドメインに割り当てるホスト ID を指定します。
- `<gprop:GenericPropertykey="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:GenericPropertykey="progress">` タグ。コマンドの進捗の割合を指定します。
- `<gprop:GenericPropertykey="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:GenericPropertykey="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。

CPU (cpu) リソース

`add-vcpu`、`set-vcpu`、および `remove-vcpu` XML 要求処理と同等の機能で、次のように `<gpropGenericProperty key="wcore">` タグの値を設定します。

- `-c` オプションを使用する場合、指定された全体のコア数を `wcore` プロパティに設定します。
- `-c` オプションを使用しない場合、`0` を `wcore` プロパティに設定します。

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 18-7 cpu の XML の例

次の例は、`ldm add-vcpu -c 1 ldg1` コマンドと同等の XML 要求を示しています。

```
<?xml version="1.0"?>
<LDM_interface version="1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding">
```

例 18-7 cpu の XML の例 (続き)

```

<cmd>
  <action>add-vcpu</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>cpu</rasd:OtherResourceType>
            <rasd:AllocationUnits>8</rasd:AllocationUnits>
            <gprop:GenericProperty key="wcore">1</gprop:GenericProperty>
          </Item>
        </Section>
      </Content>
    </Envelope>
  </data>
</cmd>
</LDM_interface>

```

例 18-8 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、list-bindings サブ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/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
          <Section xsi:type="ovf:ResourceAllocationSection_Type">
            <Item>
              <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
              <uuid>1e04cdbc-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>
              <threading>8</threading>
              <cpu-arch>native</cpu-arch>
              <rc-add-policy/>
              <gprop:GenericProperty key="state">active</gprop:GenericProperty>
            </Item>
          </Section>

```

例 18-8 ldm list-bindings コマンドによる cpu XML セクションの出力 (続き)

```

<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 18-9 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、list-domain サブ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-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>

```

例 18-9 ldm list-domain コマンドによる cpu XML セクションの出力 (続き)

```
</cmd>
</LDM_interface>
```

MAU (mau) リソース

注 - 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 (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 18-10 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>
```

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。

メモリー (memory) リソース

例 18-11 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>
```

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。

仮想ディスクサーバー (vds) リソース

例 18-12 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 (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。

仮想ディスクサーバーボリューム (vds_volume) リソース

例 18-13 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>
```

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

ディスク (disk) リソース

例 18-14 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>
```

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` – この仮想ディスクを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。また、タイムアウトによって、いずれかの `vds` への接続が指定の時間内に確実に行われます。

仮想スイッチ (vsw) リソース

例 18-15 vsw の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg2">
```

例 18-15 vsw の XML の例 (続き)

```

<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>

```

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。このリソースに含まれる必要のある <gprop:GenericProperty> には、service_name キーが付いており、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ステータスの更新を取得しません。デフォルトでは、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せん。
- 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。

ネットワーク (network) リソース

例 18-16 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>
```

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ステータスの更新を取得しません。デフォルトでは、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せん。
- `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 – SR-IOV が推奨されるため NIU ハイブリッド I/O 機能は非推奨となっています。

仮想コンソール端末集配信装置 (vcc) リソース

例 18-17 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>
```

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

変数 (var) リソース

例 18-18 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>
        <gprop:GenericProperty key="value">test1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

例 18-18 var の XML の例 (続き)

```

    </Item>
  </Section>
</Content>
</Envelope>

```

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

物理 I/O デバイス (physio_device) リソース

例 18-19 physio_device の XML の例

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">pci@780</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

physio_device 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、<gprop:GenericProperty> タグのみです。このタグには、"name" というキープロパティ値があり、記述される I/O デバイスの名前が含まれています。

SP 構成 (spconfig) リソース

例 18-20 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>

```

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用されます。

DRM ポリシー構成 (policy) リソース

例 18-21 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>
      <gprop:GenericProperty key="policy_attack">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_decay">9</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

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

仮想データプレーンチャネルサービス (vdpcs) リソース

例 18-22 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>
```

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ネルサービス (vdpcs) リソースの名前が含まれています。

仮想データプレーンチャネルクライアント (vdpcc) リソース

例 18-23 vdpcc の XML の例

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vdpcc</rasd:OtherResourceType>
      </Item>
    </Section>
  </Content>
</Envelope>
```

例 18-23 vdpcc の XML の例 (続き)

```

    <gprop:GenericProperty key="vdpcc_name">vdpcc</gprop:GenericProperty>
    <gprop:GenericProperty
      key="service_name">ldg1-vdpcs</gprop:GenericProperty>
  </Item>
</Section>
</Content>
</Envelope>

```

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。

- vdpcc_name – 仮想データプレーンチャンネルクライアント (vdpcc) の名前
- service_name – この vdpcc 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

コンソール (console) リソース

例 18-24 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>
      </Item>
    </Section>
  </Content>
</Envelope>

```

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

ドメインの移行

次の例は、`migrate-domain` サブコマンドの `<data>` セクションの内容を示しています。

例 18-25 `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>
```

ここでは:

- 1 番目の `<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 番目の `<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す。SASL 64 ビットでエンコードする必要があります。
 - `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す。SASL 64 ビットでエンコードする必要があります。

注 - Logical Domains Manager では、`sasl_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、`sasl_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。

XML スキーマ

次のリストには、/opt/SUNWldm/bin/schemas ディレクトリの各 XML スキーマファイル名が表示されています。次のスキーマは Logical Domains Manager で使用されません。

- 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

この一覧では、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)
auditreduce	監査証跡ファイルの監査レコードのマージと選択 (auditreduce(1M) のマニュアルページを参照)。
Auditing (監査)	Oracle Solaris OS 監査を使用してセキュリティーの変更元を識別すること
Authorization (承認)	Oracle Solaris OS RBAC を使用して承認を設定すること

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 の有効化 (bsmconv(1M) のマニュアルページを参照)。
bsmunconv	BSM の無効化 (bsmunconv(1M) のマニュアルページを参照)。

C

CD	Compact Disc (コンパクトディスク)
-----------	--------------------------

CLI	コマンド行インタフェース
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
CPU	Central Processing Unit (中央演算処理装置)
CWQ	Control Word Queue (暗号化装置)
D	
DHCP	動的ホスト構成プロトコル
DIO	Direct I/O
DMA	Direct Memory Access (ダイレクトメモリーアクセス)。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 (Veritas)
domain	Logical Domain (論理ドメイン) を参照してください。
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
DPS	Data Plane Software
DR	Dynamic Reconfiguration (動的再構成)
drd	Logical Domains Manager の Oracle Solaris 10 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	動的なリソース管理
DS	Domain Service module (ドメインサービスモジュール)(Oracle Solaris 10 OS)
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 (Oracle Solaris 10 OS)

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 10 OS 障害管理デーモン (fmd(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ー (format(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の生成 (fmthard(1M) のマニュアルページを参照)。
FTP	File Transfer Protocol (ファイル転送プロトコル)

G

Gb	Gigabit (ギガビット)
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。
GLDv3	Generic LAN Driver version 3 (汎用 LAN ドライバ version 3)

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
HDD	Hard Disk Drive (ハードディスクドライブ)
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 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
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
ioctl	input/output control call (I/O 制御コール)
IP	Internet Protocol (インターネットプロトコル)
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)
ISO	International Organization for Standardization (国際標準化機構)

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。
LUN	Logical Unit Number (論理ユニット番号)
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 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) の略
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 (see the metadb(1M) のマニュアルページを参照)。
metaset	ディスクセットの構成 (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 (mhd(7i) のマニュアルページを参照)。
MIB	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	
NAT	Network Address Translation (ネットワークアドレス変換)
ndpsldcc	Netra DPS Logical Domain Channel Client。「vdpc」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。
NFS	Network File System (ネットワークファイルシステム)
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
OS	オペレーティングシステム
OVF	Open Virtualization Format

P

P2V	Logical Domains 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
PCIe	PCI Express バス
PCI-X	PCI 拡張バス
pcpu	物理 CPU
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の出力 (praudit(1M) のマニュアルページを参照)。
PRI	Priority (優先度)
PROM	Programmable Read-Only Memory (プログラム可能な読み取り専用メモリー)

R

RA	Real Address (実アドレス)
RAID	Redundant Array of Inexpensive Disks
RBAC	Role-Based Access Control (役割に基づくアクセス制御)
RPC	Remote Procedure Call (リモート手続き呼び出し)

S

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。
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。
SCSI	Small Computer System Interface
Service Domain (サービルドメイン)	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
SMA	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)
SNMP	SNMP
Service Processor (SP、サービスプロセッサ)	システムコントローラ (SC) と呼ばれる SP は、物理マシンをモニターし実行します。

SR-IOV	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)。

T

TCP	Transmission Control Protocol (伝送制御プロトコル)
TLS	Transport Layer Security

U

UDP	User Datagram Protocol (ユーザーダイアグラムプロトコル)
UFS	UNIX File System (UNIX ファイルシステム)
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
USB	Universal Serial Bus (ユニバーサルシリアルバス)
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ンネルクライアント
vdpcs	Netra DPS 環境における仮想データプレーンチャンネルサービス
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。
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せん。
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。
VLAN	Virtual Local Area Network (仮想ローカルエリアネットワーク)
vldc	Virtual Logical Domain Channel Service (仮想論理ドメインチャンネルサービス)
vldcc	Virtual Logical Domain Channel Client (仮想論理ドメインチャンネルクライアント)
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。
vNTS	Virtual Network Terminal Service (仮想ネットワーク端末サービス)
vntsd	Logical Domains コンソールの Oracle Solaris 10 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

W

WAN Wide-Area Network (広域ネットワーク)

X

XFP eXtreme Fast Path

XML eXtensible Markup Language

XMPP eXtensible Messaging and Presence Protocol

Z

ZFS Zettabyte File System (Oracle Solaris 10 OS)

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

C

cancel-reconf サブコマンド, 210
CLI, 「コマンド行インタフェース」を参照
CPUクロックサイクルのスキップ, 292
CPUコアの無効化, 292

D

DR, 「動的再構成」を参照

I

I/O ドメイン, 81-82, 83-88, 89-100, 100-117
 PCIe SR-IOV 仮想機能の使用, 100-117
 PCI EXPRESS (PCIe) バス, 81-82
 移行の制限, 82
 作成, 84
 割り当て
 PCIe バス, 83-88
 エンドポイントデバイス, 89-100
I/O ドメインの作成, PCIe バス全体, 84

L

LDC, 「論理ドメインチャネル」を参照
ldm(1M) コマンド, 26
ldm(1M) マニュアルページ, 26
ldmconfig(1M) コマンド, 28, 285, 286
ldmd, Logical Domains Manager デーモン, 26
ldmp2v(1M) コマンド, 272

ldm サブコマンド

 cancel-reconf, 210
 ユーザー承認, 51
ldmCoreTable テーブル, 327-328
ldmCreate トラップ, 331
ldmCryptoTable テーブル, 326-327
ldmDestroy トラップ, 331-332
ldmEnvVarsTable テーブル, 311
ldmIOBusTable テーブル, 327
ldmPolicyTable テーブル, 311-312
ldmSPConfigTable テーブル, 312-313
ldmStateChange トラップ, 332
ldmTable テーブル, 308-311
ldmVccChange トラップ, 335-336
ldmVccTable テーブル, 324-325
ldmVconsChange トラップ, 336
ldmVconsTable テーブル, 325
ldmVconsVccRelTable テーブル, 325
ldmVCpuChange トラップ, 332
ldmVcpuTable テーブル, 314-316
ldmVdiskChange トラップ, 334
ldmVdiskTable テーブル, 319-320
ldmVdsChange トラップ, 333-334
ldmVdsdevTable テーブル, 318-319
ldmVdsTable テーブル, 318
ldmVMemChange トラップ, 332-333
ldmVmemPhysBindTable テーブル, 317
ldmVmemTable テーブル, 316-317
ldmVnetChange トラップ, 335
ldmVnetTable テーブル, 323-324
ldmVswChange トラップ, 334-335
ldmVswTable テーブル, 322-323

Logical Domains Manager, 22, 24
XMLスキーマの使用, 345
検出メカニズム, 341
デーモン (ldmd), 26
と Oracle VM Server for SPARC MIB, 298

M

MIB, 295
Oracle VM Server for SPARC MIB オブジェクト, 取得, 306-307
Oracle VM Server for SPARC MIB ソフトウェア
インストール, 300-303
構成, 300-303
削除, 300-303
Oracle VM Server for SPARC MIB テーブル
CPU リソースプールのスカラー変数, 313
I/O バステーブル (ldomIOBusTable), 327
I/O バスリソースプールのスカラー変数, 314
Logical Domains バージョン情報のスカラー変数, 328
暗号化装置テーブル
(ldomCryptoTable), 326-327
暗号化リソースプールのスカラー変数, 314
仮想 CPU テーブル (ldomVcpuTable), 314-316
仮想コンソール関係テーブル
(ldomVconsVccRelTable), 325
仮想コンソールグループテーブル
(ldomVconsTable), 325
仮想コンソール端末集配信装置テーブル
(ldomVccTable), 324-325
仮想スイッチサービスデバイステーブル
(ldomVswTable), 322-323
仮想ディスクサービステーブル
(ldomVdsTable), 318
仮想ディスクサービスデバイステーブル
(ldomVdsdevTable), 318-319
仮想ディスクテーブル
(ldomVdiskTable), 319-320
仮想ネットワークデバイステーブル
(ldomVnetTable), 323-324
仮想メモリーテーブル
(ldomVmemTable), 316-317

Oracle VM Server for SPARC MIB テーブル (続き)
仮想メモリーの物理バインディングテーブル
(ldomVmemPhysBindTable), 317
環境変数テーブル (ldomEnvVarsTable), 311
コアテーブル (ldomCoreTable), 327-328
サービスプロセッサ構成テーブル
(ldomSPConfigTable), 312-313
ドメインテーブル (ldomTable), 308-311
ドメインポリシーテーブル
(ldomPolicyTable), 311-312
Oracle VM Server for SPARC MIB トラップ, 331-336
仮想 CPU の変更 (ldomVcpuChange), 332
仮想コンソールグループの変更
(ldomVconsChange), 336
仮想コンソール端末集配信装置の変更
(ldomVccChange), 335-336
仮想スイッチの変更 (ldomVswChange), 334-335
仮想ディスクサービスの更改
(ldomVdsChange), 333-334
仮想ディスクの更改 (ldomVdiskChange), 334
仮想ネットワークの更改 (ldomVnetChange), 335
仮想メモリーの更改 (ldomVmemChange), 332-333
受信, 330-331
送信, 329-330
ドメインの削除 (ldomDestroy), 331-332
ドメインの作成 (ldomCreate), 331
ドメインの状態変更 (ldomStateChange), 332
Oracle VM Server for SPARC MIB モジュール,
ロード, 300-303

O

Oracle VM Server for SPARC MIB
オブジェクトツリー, 298-299
ソフトウェアコンポーネント, 296
と Logical Domains Manager, 298

P

PCI EXPRESS (PCIe) バス, 81-82
primary ドメイン, 25
primary ドメイン, リポート, 93-94
primary ドメインのリポート, 93-94

S

snmpv3 ユーザー, 作成, 303-304
 Solaris power aware dispatcher (PAD), 293
 SUNWldm パッケージ, 26

V

vNTS, 324-325

X

XML スキーマ, Logical Domains Manager での使用, 345
 XML ベースの制御インタフェース, 解析, 298

い

移行, 対話型でない, 207
 移行の制限, I/O ドメイン, 82
 インストール, Oracle VM Server for SPARC MIB ソフトウェア, 300-303

か

解析, XML ベースの制御インタフェース, 298
 仮想ディスクマルチパス, 134
 仮想デバイス
 I/O, 26
 仮想コンソール端末集配信装置 (vcc), 27
 仮想スイッチ (vsw), 26
 仮想ディスククライアント (vdc), 27
 仮想ディスクサービス (vds), 27
 仮想ネットワーク (vnet), 26
 仮想ネットワーク端末サーバデーモン (vntsd), 27
 仮想マシン, 24
 環境変数, 設定, 305
 管理情報ベース (MIB), 295

き

起動, ドメイン, 337-338

け

計画
 直接 I/O (DIO), 92
 直接 I/O (Direct I/O、DIO), 92
 ゲストドメイン, 25

こ

構成
 Oracle VM Server for SPARC MIB ソフトウェア, 300-303
 ジャンボフレーム, 188-193
 ブートの選択, 27
 コマンド
 ldm(1M), 26
 ldmconfig(1M), 28, 285, 286
 ldmp2v(1M), 272
 コマンド行インタフェース, 26

さ

サービスドメイン, 25, 26
 サービスプロセッサ (SP), 物理マシンのモニタリングと実行, 24
 削除, Oracle VM Server for SPARC MIB ソフトウェア, 300-303
 作成, snmpv3 ユーザー, 303-304

し

システム管理エージェント, 297
 システムコントローラ, 「サービスプロセッサ (SP)」を参照
 ジャンボフレーム, 構成, 188-193
 受信, Oracle VM Server for SPARC MIB トラップ, 330-331

取得

Oracle VM Server for SPARC MIB オブジェクト, 306-307

Oracle VM Server for SPARC MIB 情報, 308-328

使用

I/O ドメインで PCIe SR-IOV 仮想機能を, 100-117

リンクベースの IPMP, 173-177

障害と復旧の情報の提供, 提供, 298

承認, `ldm` サブコマンド, 51

せ

制御ドメイン, 25

設定

環境変数, 305

電力の制限, 292

そ

送信, Oracle VM Server for SPARC MIB トラップ, 329-330

た

対話型でないドメインの移行, 207

ち

遅延再構成, 210

直接 I/O (Direct I/O、DIO), 計画, 92

て

提供

障害と復旧の情報の提供, 298

トラップ, 298

停止, ドメイン, 338-339

テーブル, 「Oracle VM Server for SPARC MIB テーブル」を参照

デーモン

`drd`, 210

`ldmd`, 26

`vntsd`, 27

電源管理 (PM), 292, 293

電力の制限, 292

と

動的再構成 (DR), 209

動的再構成デーモン (`drd`), 210

ドメイン

起動, 337-338

サービス, 26

停止, 338-339

の種類, 25

のタイプ, 25

ドメインの移行, 対話型でない, 207

トラップ

「Oracle VM Server for SPARC MIB トラップ」を参照

提供, 298

は

ハイパーバイザ, 22

パッケージ, `SUNwLdm`, 26

ふ

物理デバイス, 25, 26

物理マシン, 24

ま

マルチパス, 仮想ディスク, 134

め

メモリー電源管理 (PM), 292

メモリーの動的再構成 (DR), 226

や

役割, 論理ドメイン, 25

り

リソース

「仮想デバイス」も参照

定義, 23

リンクベースの IPMP, 使用, 173-177

る

ルートドメイン, 25

ろ

ロード, Oracle VM Server for SPARC MIB モ
ジュール, 300-303

論理ドメイン

定義, 22

役割, 25

論理ドメインチャンネル (LDC), 24

わ

割り当て

I/O ドメインへの PCIe, 83-88

I/O ドメインへのエンドポイントデバイ
ス, 89-100

